1983

A digital filter representation of the ASQ-81 magnetometer.

Huete, Michael Charles.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/19752

Dudl
Mont

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DIGITAL FILTER REPRESENTATION OF THE ASQ-81 MAGNETOMETER

by

Michael Charles Huete

September 1983

Thesis Advisor: Andrew R. Ochadlick

Approved for public release, distribution unlimited

REPORT DOCUMENTATION PAGE — READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
A Digital Filter Representation of the ASQ-81 Magnetometer

5. TYPE OF REPORT & PERIOD COVERED
Master's Thesis
September 1983

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Michael Charles Huete

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Postgraduate School
Monterey, Ca. 93943

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Naval Postgraduate School
Monterey, Ca. 93943

12. REPORT DATE
September 1983

13. NUMBER OF PAGES
121

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release, distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Digital Filter, AN/ASQ-81, Magnetometer, Magnetic Anomaly Detection, MAD, Bilinear Transformation, FORTRAN, Geomagnetic Noise, Geomagnetic Field

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
A digital filter representation of the ASQ-81 magnetometer is derived from the s-plane transfer functions of the system through the use of a bilinear transformation. A FORTRAN computer program is written which applies this representation to time-sampled total magnetic field data in order to obtain a time series representation of ASQ-81 filtered total field. A series of simulations and a field experiment are conducted which verify the program output. Applications of this program include usage in conjunction with geomagnetic field

data in order to produce a new data set representative of geomagnetic noise observed by Navy MAD (Magnetic Anomaly Detection) aircraft with the potential to investigate techniques of reducing geomagnetic noise in MAD aircraft.

A Digital Filter Representation of the ASQ-81 Magnetometer

by

Michael Charles Huete

Lieutenant, United States Navy

B. S. E. E., Tulane University, 1976

Submitted in partial fulfillment of the

requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY
(Antisubmarine Warfare)

from the

NAVAL POSTGRADUATE SCHOOL

September 1983

ABSTRACT

A digital filter representation of the ASQ-81 magnetometer is derived from the s-plane transfer functions of the system through the use of a bilinear transformation. A FORTRAN computer program is written which applies this representation to time-sampled total magnetic field data in order to obtain a time series representation of ASQ-81 filtered total field. A series of simulations and a field experiment are conducted which verify the program output. Applications of this program include usage in conjunction with geomagnetic field data in order to produce a new data set representative of geomagnetic noise observed by Navy MAD (Magnetic Anomaly Detection) aircraft with the potential to investigate techniques of reducing geomagnetic noise in MAD aircraft.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

The detection and location of submarines (and other magnetic bodies) through the discrimination of changes or anomalies in the Earth's magnetic field is called Magnetic Anomaly Detection or MAD. In this technique, a magnetometer measures the magnitude of the Earth's magnetic field and provides an indication of that magnitude, or, more usually, an indication of changes in the magnitude of the Earth's field. These changes, or anomalies, can indicate the presence of magnetized bodies which may or may not be a submarine.

The magnetometer currently in use in the United States Navy for use in this MAD process is the AN/ASQ-81 metastable helium vapor total field magnetometer.

Research is currently being conducted at the Naval Postgraduate School in Monterey, California, in various aspects of the applications of magnetometers, including Magnetic Anomaly Detection (MAD). Within the context of this research, magnetic field measurements are made through the use of sets of wire wound coils vice any specific magnetometer or magnetic detecting system. The data collected through the use of these coils is evaluated and

processed in a variety of methods for different project goals.

This thesis project is designed to produce an acceptable alternative to the physical presence of an experimental AN/ASQ-81 magnetometer at the postgraduate school by allowing the determination, in conjunction with other research in progress, of the output of the AN/ASQ-81 magnetometer from the data collected from the school's measurement coils. It is hoped that this will assist future research projects as, for example, in allowing a determination of environmental noise of such characteristics as to affect the AN/ASQ-81 magnetometer operationally with the eventual goal of providing an environmental noise index or a system of removing such noise from the magnetometer-detection system.

II. GEOMAGNETICS REVIEW

A. EARTH'S MAGNETIC FIELD

1. Constituents of the Geomagnetic Field

The most common method of specifying the consituting parts of the geomagnetic field is to divide the field in terms of distance from the center of the Earth. This method results in three classifications: internal, crustal, and external. [Ref. 1]

The internal field originates in the core region and is the most stable field, containing only extremely low frequency temporal variations. The crustal, or anomalous, field arises from modifications made on the internal field by materials and structures in the Earth's crust. These variations are not constant with regard to spatial locations, and comprise part of what is known as geological variations. The external field is the most dynamic and arises from many sources, including the interaction between the solar wind and the Earth's magnetic field.

In addition to this method of defining the Earth's magnetic field is the method of time variations. This method consists of considering that part of the field which varies with periodicities greater than about one year as the

steady field and everything else as the variation field. [Ref. 2]

The steady field consists of the internal field, also refered to as the main field. Slow variations of the main field with periods of years or longer are referred to as secular variations.

There are various elements that contribute to the geomagnetic field, some of which are external to the Earth's surface. External contributions make up only a small part of the steady field, but play an important role in the variation field. These external sources include current systems in the Earth's upper atmosphere affected by solar electromagnetic radiation and gravitation, solar corpuscular radiation and the interaction of solar plasma with the main field, and the effect of the solar inteplanetary field. [Ref. 3]

The geomagnetic field changes with time. As previously mentioned, very slow variations in the main field with periods of on the order of years to thousands of years are referred to as secular variations. Secular variations are caused by a variation in the strength or orientation of the Earth's center dipole.

Other time variations of the field can be categorized into quiet variation fields and disturbed variation fields. Disturbed variation fields include geomagnetic micropulsations, which are of particular interest to

operational forces as these can mask target signatures and are therefore a source of noise to MAD sensors.

Quiet variation fields are those which are not due to disturbances in the interplanetary environment and which vary slowly and regularly. [Ref. 3]

Disturbed variation fields are geomagnetic field variations that appear to be the result of interplanetary environmental changes and do not posess a simple periodicity. These variations include ionospheric disturbances, the aurora, geomagnetic storms, and geomagnetic micropulsations.

2. Elements of the Magnetic Field Vector

The geomagnetic field vector is characterized at any point by its direction and magnitude. This is commonly accomplished through a system of coordinates as shown in Figure 2.1. The field is measured in terms of local coordinates with respect to true North. [Ref. 3]

The various coordinates are referred to as magnetic elements and are defined as follows:

B: Total field intensity (the symbol F is sometimes also used, as in this figure.)

H: Horizontal component

X: Northward, or NorthSouth component

Y: Eastward, or EastWest component

Z: Downward, or Vertical component

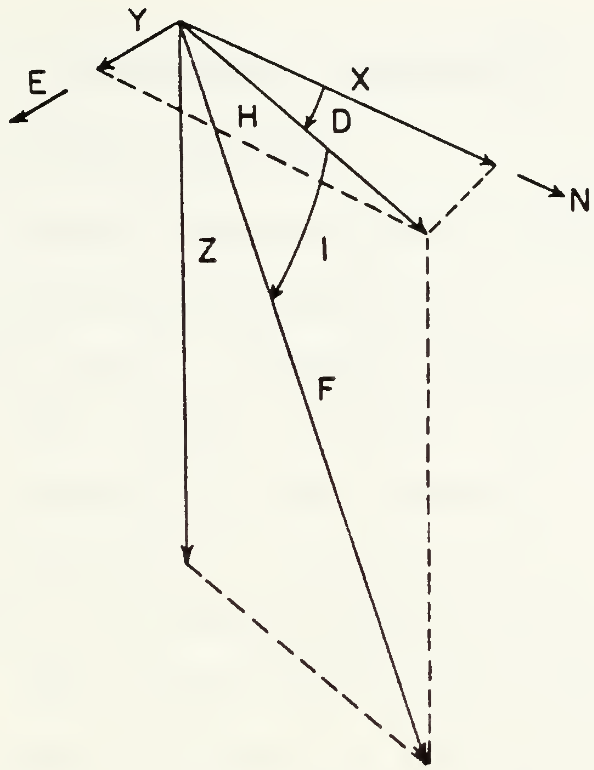

Figure 2.1 Magnetic Field Elements [Ref. 4].

<u>D</u>: Declination or magnetic variation

This is the angle between <u>X</u> and <u>H</u> and is measured positive eastward.

<u>I</u> Inclination or Dip Angle.

This is the angle between <u>H</u> and <u>B</u> (or <u>F</u>) and is measured positive downward.

III. THE AN/ASQ-81 MAGNETOMETER

A. DESCRIPTION OF SYSTEM OPERATION

The Magnetic Anomaly Detecting set currently in use in the U S Navy is the AN/ASQ-81 magnetometer. This set is used to locate and classify submerged submarines by sensing disturbances in the Earth's magnetic field (anomalies) caused by the presence of the magnetic mass of the submarine. The disturbance of the Earth's field is detected by the magnetometer, processed through filtering circuits, and amplified. The output signal of the magnetometer is displayed on a chart recorder for interpretation by an operator.

The magnetic detecting set is a metastable helium vapor magnetometer. The operation of the magnetometer is based on the light absorbtion properties of helium gas subjected to certain light stimulus (optical pumping), radio frequency excitation, and the Earth's magnetic field. The magnetometer consists of a helium lamp, lens and polarizer to generate a beam of polarized light radiation. This focused and polarized light beam is directed through a helium absorbtion cell to an infrared (IR) detector. Some of the helium gas in the absorbtion cell is maintained in a metastable state by application of VHF excitation.

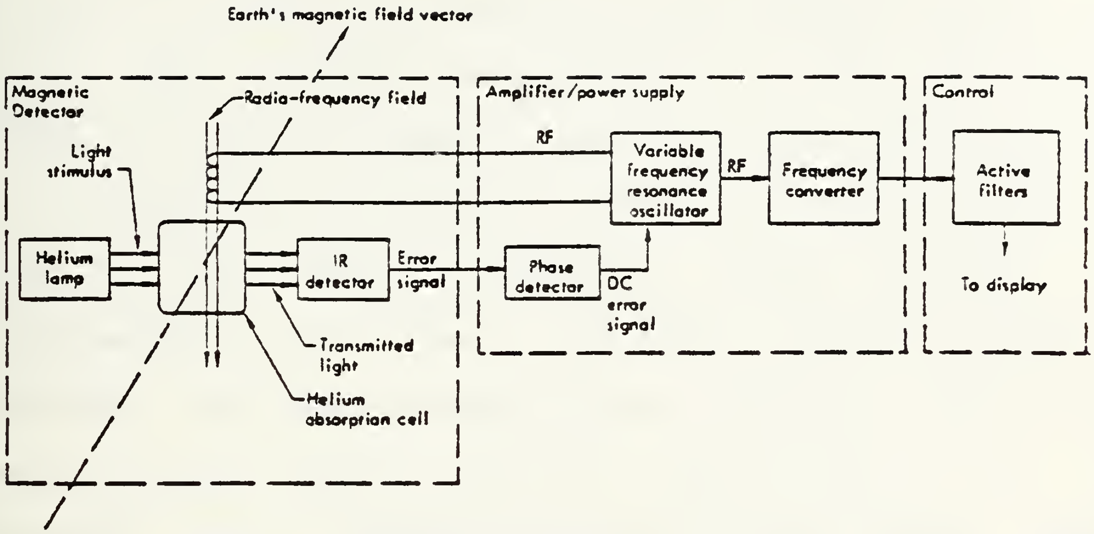

Figure 3.1 : Metastable Helium Magnetometer [Ref. 5].

The Earth's magnetic field imposes a magnetic force upon the excited helium vapor atoms to force the atoms into one of three energy sublevels. This is called the Zeeman effect. The rate or frequency of atomic precession caused by this effect is called the Larmor frequency. A helium lamp is used to optically pump the atoms in the absorbtion cell, with the result that the polarized light energy passing through the absorbtion cell will polarize (magnetize) the helium atoms in the absorbtion cells by selectively pumping the Zeeman levels of the energy of the helium atoms in the cell. The magnetization direction is determined by the polarization of the photons from the helium lamp.

RF energy is then introduced to the absorbtion cell in the form of an additional magnetic field imposed through the use of coils oriented perpendicular to the precessed polarized helium atoms in the absorbtion cell and energized by a variable frequency RF oscillator. The RF oscillator is tuned to the Larmor frequency, which results in depolarization of the atoms. The atoms attempt to equally repopulate the Zeeman energy levels. However, the helium lamp is still beaming polarized light energy into the absorbtion cell, causing the atoms to absorb light energy and rise to an excited energy levbel. This absorbtion of light energy is detected through the use of an infrared detector. The RF oscillator frequency producing maximum light absorbtion is called the resonant frequency, and is determined through the use of a servo loop from the infrared detector to the RF variable frequency oscillator.

Therefore, any change in the Earth's magnetic field intensity will result in a change in the Larmor frequency of the helium atoms in the helium absorbtion cell. This new Larmor frequency will be detected by the ASQ-81 magnetometer. Since the gyromagnetic ratio of helium is 28.024 HZ per gamma, this detection of the resonant frequency provides a measurement of the Earth's magnetic field intensity at any given time. A change in the Earth's

magnetic field intensity could signal the presence of a submerged submarine.

The output resonant frequency developed by the magnetometer is converted to a proportional output voltage which is filtered through the Magnetic Anomaly Detection (MAD) bandpass filters for environmental noise reduction and utilized to drive a chart recorder for observation by an operator. [Ref. 6]

B. TRANSFER FUNCTIONS

Transfer functions for the AN/ASQ-81 filters were obtained from the manufacturer of the AN/ASQ-81 detecting set, Texas Instruments of Dallas, Texas. These transfer functions are listed in Appendix A and are in the form of H(s), that is, the frequency domain, or S domain, where S = jw. The s-domain representation for transfer functions is routinely utilized to express output system characteristics for given system inputs. As the S domain representation is not utilzed further in this discussion, it will not be further explained.

As the output signal of the ASQ-81 magnetometer is filtered through a fixed high-pass system, then through a selectable low pass system and a selectable high pass system (as shown in Figure 3.2 below), the transfer funtions are listed in this order.

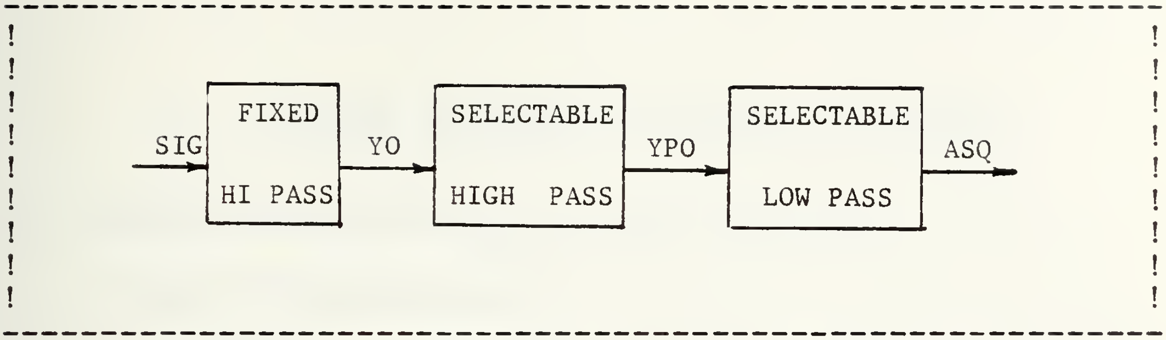

Figure 3.2 : Signal Flow Diagram for ASQ-81

Further discussion will be made of the selectable filters later.

IV. DIGITAL FILTERING MODELLING OF SYSTEMS

A. SEQUENCE REPRESENTATION OF TIME FUNCTIONS

1. Signal Representation

A signal can be defined as a function that conveys information, generally about the state or behavior of a physical system. Although signals can be represented in many ways, the information conveyed by the signal is contained in a pattern of variations of some form. Signals are represented mathematically as functions of one or more independent variables, one of the most common of which is time.

The independent variable of the mathematical representation of a signal may be continuous or discrete. Continuous time signals are signals that are defined over continualy values of time and are therefore represented by continuous-variabled functions. Discrete time signals are defined at discrete time intervals and are therefore represented by functions whose independent variable(s) take on discrete values only. Discrete-time signals are represented as sequences of numbers. [Ref. 7]

In addition to the fact that the independent variables can be either continuous or discrete, the signal amplitude can be either continuous or discrete. Digital

signals are those for which both time and amplitude are discrete. Analog signals are those for which both time and amplitude are continuous.

Digital signal processing deals with transformations of signals that are discrete in both time and amplitude, usually represented by sequences of numbers. The nth number in the sequence x being processed is usually represented as x(n), and is formally written as:

$$x=[x(n)], \quad -\infty < n < +\infty$$

In general, an arbitrary sequence can be expressed as

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\ d(n-k)$$

where d(n-k) is the unit sample at time k. In other words, an arbitrary sequence may be expressed as a sum of scaled, shifted unit samples, where the scaling factor is equal to the amplitude of the sequence at that time.

2. Linear Shift-Invariant Systems

A system is defined mathematically as a unique transformation or operator that maps an input sequence [x(n)] into an output sequence [y(n)]. This is denoted as:

$$y(n) = T[x(n)]$$

and is often depicted as in Figure 4.1.

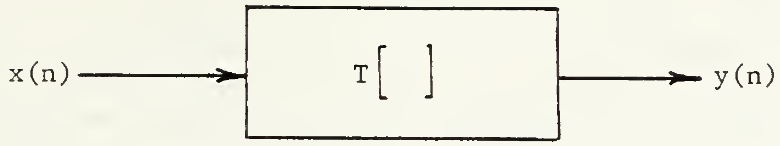

Figure 4.1: Representation of Transformation of an Input Sequence to an Output Sequence. [Ref. 7]

Classes of discrete time systems are defined by placing constraints on the transformation T[].

The class of <u>linear systems</u> is defined by the principle of superposition. If $y_1(n)$ and $y_2(n)$ are the responses when $x_1(n)$ and $x_2(n)$ are the inputs, then a system is linear if

$$T[ax_1(n) + bx_2(n)] = aT[x_1(n)] + bT[x_2(n)]$$
$$= ay_1(n) + by_2(n)$$

for any arbitrary constants a and b. This, together with the concept of representing a sequence by a sum of delayed and scaled unit-sample sequences, suggests that a linear system can be characterized by its unit-sample response. Specifically, let $h_k(n)$ be the response of the system to d(n-k), a unit sample occurring at n=k. Then

$$y(n) = T[\sum_{k=-\infty}^{\infty} x(k)\, d(n-k)] \qquad \text{or,}$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, T[d(n-k)] = \sum_{k=-\infty}^{\infty} x(k)\, h_k(n)$$

Thus the system response can be expressed in terms of the response of the system to d(n-k).

The class of <u>shift invariant systems</u> is characterized by the property that if y(n) is the response to x(n), then y(n-k) is the response to x(n-k), where k is a positive or negative integer. When the index n is associated with time, shift-invariance corresponds to <u>time-invariance</u>. The property of shift invariance implies that if h(n) is the response to d(n), then the response to d(n-k) is simply h(n-k). Therefore

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k)$$

and any linear shift-invariant system is completely characterized by its unit-sample response h(n).

A subclass of linear shift-invariant systems consists of those systems for which the input x(n) and the output y(n) satisfy an Nth-order linear constant-coefficient difference equation of the form

$$\sum_{k=0}^{N} a_k\, y(n-k) = \sum_{r=0}^{M} b_r\, x(n-r)$$

If the assumption of causality is made about the system, a linear difference equation provides an explicit relationship between the input to the system and the output of the system. This can be seen by rewriting the previous equation as

$$y(n) = \sum_{k=1}^{N} c_k\, y(n-k) + \sum_{r=0}^{M} d_r\, x(n-r)$$

where $c_k = -a_k / a_0$ and $c_r = b_r / a_0$.

Thus the nth value of the output can be computed from the nth value of the input and the N and M past values of the output and input, respectively. The difference equation not only represents the system for theoretical purposes, but it may also serve as a computational realization of the system. The z-Transform makes use of this property to realize systems.

B. THE z-TRANSFORM

1. Description of the z-Transform

The <u>z-transform</u> plays an important role in the analysis and representation of discrete-time linear shift-invariant systems. The z-transform, X(z), of a sequence x(n) is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

where z is a complex variable. This representation of the z-transform is referred to as the two-sided z transform. The one sided z-transform consists of the same summation for terms of n greater than or equal to zero. For the case that $x(n)=0$ for $n<0$, the one sided and two sided z transforms are equivalent.

By expressing the complex variable z in polar form as $z = re^{jw}$, the z-transform can be interpreted as the Fourier transform of x(n) multiplied by an exponential sequence. For r = 1, that is, for $|z| = 1$, the z-transform is equal to the Fourier tranform of the sequence.

2. The Bilinear Transformation

The transfer functions of analog systems are most often expressed in terms of s = jw (see section III B.). This corresponds to the analog frequency response of the system. This analog frequency response can be "mapped", that is, tranformed to the z-plane from the s-plane through the use of the bilinear transformation. The effect of utilizing the bilinear transformation is to convert a system transfer function in terms of the variable S into the system transfer function in terms of the variable z. The transformation itself is:

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

and

$$z = \frac{(2/T) + s}{(2/T) - s}$$

where T is the sampling period, that is, the time between data samples.

Thus a transform can be made from one plane to the other. In this manner, the transfer function, H(z), of a system may be obtained.

The bilinear tranformation equations may be shown to hold in general, and the use of this transformation may be shown to yield stable digital filters from stable analog filters [Ref. 7]. The bilinear transformation maps the imaginary jw axis in the s-plane onto a unit circle (of the region of convergence) in the z-plane, with the left half s-plane mapped onto the region inside the circle and the right hand (region of instability) s-plane mapped onto the region outside this circle [Ref. 8]. A complete discussion of the z-transform is available in several texts, some of which are listed in the Bibliography.

C. THE DIGITAL COMPUTATIONAL ALGORITHM

In implementing a digital filter on a digital computer such as the IBM 3033, the input-output relationship of the signals through the system being synthesized must be converted to a computational algorithm. The algorithm is specified in terms of a set of basic computations of elements. For the implementation of discrete-time systems

described by linear constant coefficient difference equations, such as the AN/ASQ-81, it is convenient to choose as these elements the basic operations of addition, delay, and multiplication by a constant. The computational algorithm for implementing the filter is then defined by a structure or network consisting of an interconnection of these basic operations. For a system transfer function of the form

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 - \sum_{k=1}^{N} a_k z^{-k}} = \frac{Y(z)}{X(z)}$$

the difference equation relating input and output is easily written down directly from the system function and is given by

$$y(n) = \sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \quad \text{[Ref. 7]}$$

This difference equation can be interpreted directly as a computational algorithm in which the delayed values of the input are multiplied by the coefficients b_k, the delayed values of the output are multiplied by the coefficients a_k, and the resulting products are added. It is now easy to see the process to be followed in obtaining the computational algorithm for the AN/ASQ-81 magnetometer

transfer function. The z-transform of the system transfer function is obtained through the use of the bilinear transformation, and is then converted into a difference equation relating input and output signals, thence to a FORTRAN computer program. A table of z-transforms of system functions is included in Appendix B.

In the FORTRAN computer program realization of the total system computational algorithm, each filter block is transformed into a separate difference equation and algorithm. This was done to enable a "building block" type approach to the program, and to minimize computational and roundoff errors.

D. THE CASCADE FORM OF THE COMPUTATIONAL ALGORITHM

Even though the direct form realization of the digital filter design may be perfectly satisfactory in a theoretical sense, it may be less than desirable in the context of realization through the use of a general purpose computer of fixed register length. The parameters of a digital filter are usually obtained with a high degree of accuracy, which results in a faithful realization of the desired system. When these parameters are quantized, as in a finite memory register within a computer, the frequency response of the resulting digital filter may differ appreciably from the original design. In fact, the quantized filter may fail to

meet design specifications although the unquantized filter does. [Ref. 7]

The sensitivity of the filter response to errors in the filter parameters is dependent upon the structure of the filter realization. Therefore, in the event of an unacceptable change in the frequency response of the filter due to quantization errors, it is often possible to minimize the effect of these errors through an alternate filter realization structure. An alternate structure to the previoulsy discussed direct form realization is the <u>cascade form</u> realization.

The direct form network structures were obtained directly from the system function H(z) written in the form of a ratio of sums. If this ratio is factored into a product of polynomials of the form

$$H(z) = A \prod_{k=1}^{[(N+1)/2]} \frac{1 + B_{1k} z^{-1} + B_{2k} z^{-2}}{1 - a_{1k} z^{-1} - a_{2k} z^{-2}}$$

this product represents a general distribution of poles and zeros and suggests a set of structures consisting of a cascade of first and second-order subsystems. There is considerable freedom in the design of the subsystems, but it is best to realize the systems using a minimum of storage.

The expression of H(z) in this form indicates the presence of poles and zeros in pairs. If poles and zeros

are not present in pairs, one of the coefficients B_{2k} or a_{2k} will be zero as appropriate. An implementation of such a cascade structure with the use of minimum memory can be obtained through a direct form II realization of each second order subsystem using techniques similar to the direct form implementation utilized previously. A cascade realization of a sixth-order system, such as the ASQ-81 system, using a direct form II realization of each second order subsystem would appear as in Figure 4.2 below.

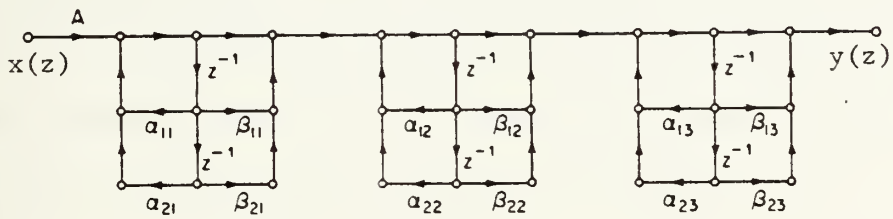

Figure 4.2: Cascade Structure With a Direct Form II Realization of Each Second Order Subsystem. [Ref. 7]

There is, theoretically, considerable flexibility in the manner in which the poles and zeros are paired together and in the order in which the resulting second-order subsystems are cascaded. However, although all such pairings and orderings are equivalent for infinite-precision arithmetic, they may differ considerably in practice owing to finite word length effects of roundoff and truncation.

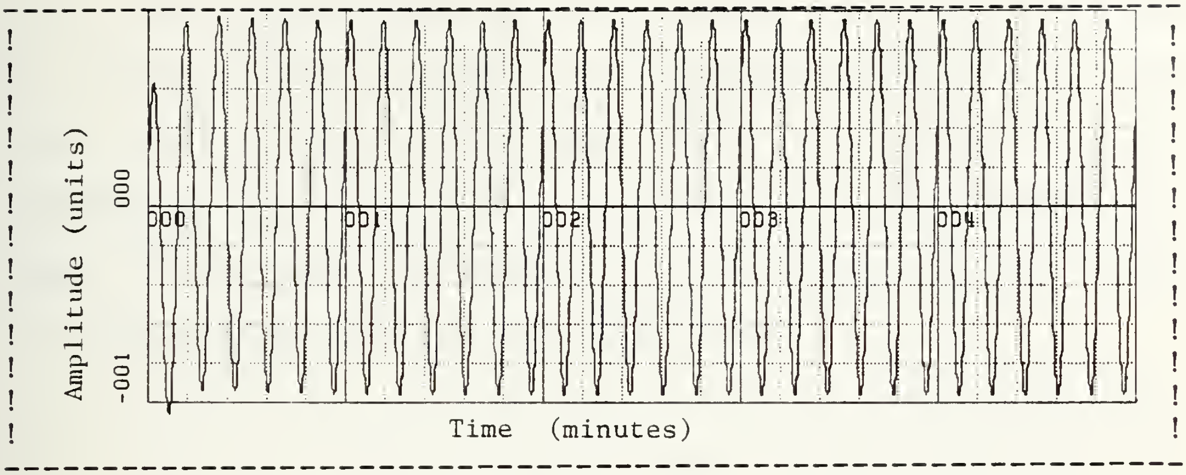

Figure 5.1: Output of First Stage Filter of Digital Filter Computer Program With Sinusoidal Input in Simulation.

Unfortunately, the second stage output of the filter showed an instability within the program design, indicated by the output of the filter being a sinusoid of increasing magnitude, as indicated in Figure 5.2 below.

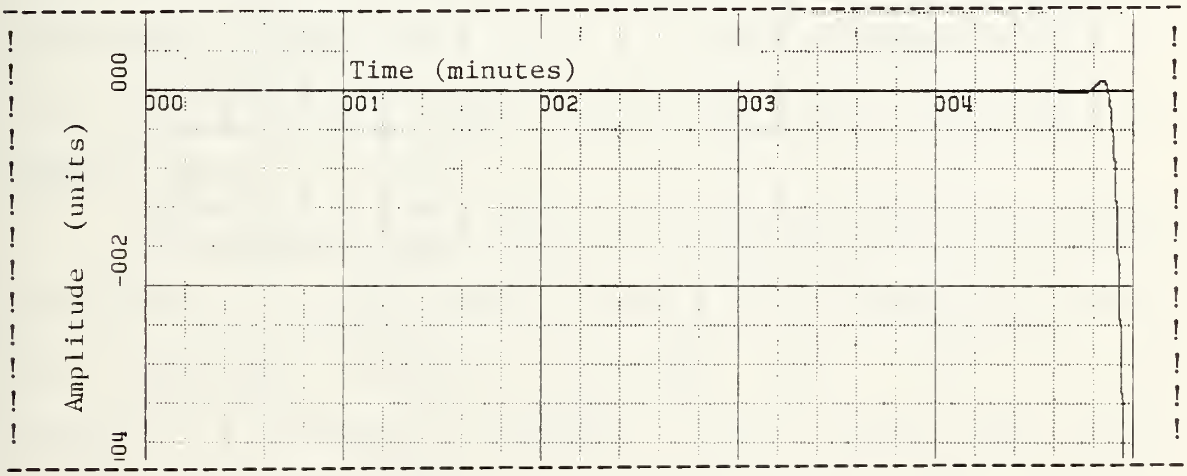

Figure 5.2: Output of Second Stage Filter Design With Input of a Sinusoid.

The stability of the third stage of the filter design was investigated by inputting the sinusoid directly to the third filter, and found to be stable. A check of the derivation of the equations, coefficients, and programming steps of the second (unstable) filter of the design failed to indicate the cause of the instability.

Computation of the poles of the z transfer function, H(z), of the second stage of the filter confirmed the instability of the design. The poles were computed to be: $0.92 \pm 0.1218i$, $1.07 \pm 0.1340i$, 0.8611, and 1.1564. Of these six poles, three lie outside the region of convergence for the z-plane, that is, within the unit circle discussed previously in Chapter IV.

The second stage of the filter was therefore redesigned using the cascade form of the direct form realization (direct form II), and tested in simulation. A copy of the software used in the simulation is enclosed in Appendix F.

The output of all three filter stages of the program were stable, as indicated in Figures 5.3 through 5.7 below. The amplitude decrease and phase shift expected were observed. The "damped overshoot" of the second stage output is due to the fact that, for values of the input function prior to time zero in the simulation, utilized in the input-output signal difference equations for the filter, the input signal was set at 0. This resulted in an instantaneous

change of the input signal from 0 to the finite value introduced in the simulation at time 0+. The "overshoot" of the filter is the filter's attempt to "match" this instantaneous jump in magnitude of the input signal. When the input signal to the filter in the simulation is zero at time zero, this overshoot effect does not occur.

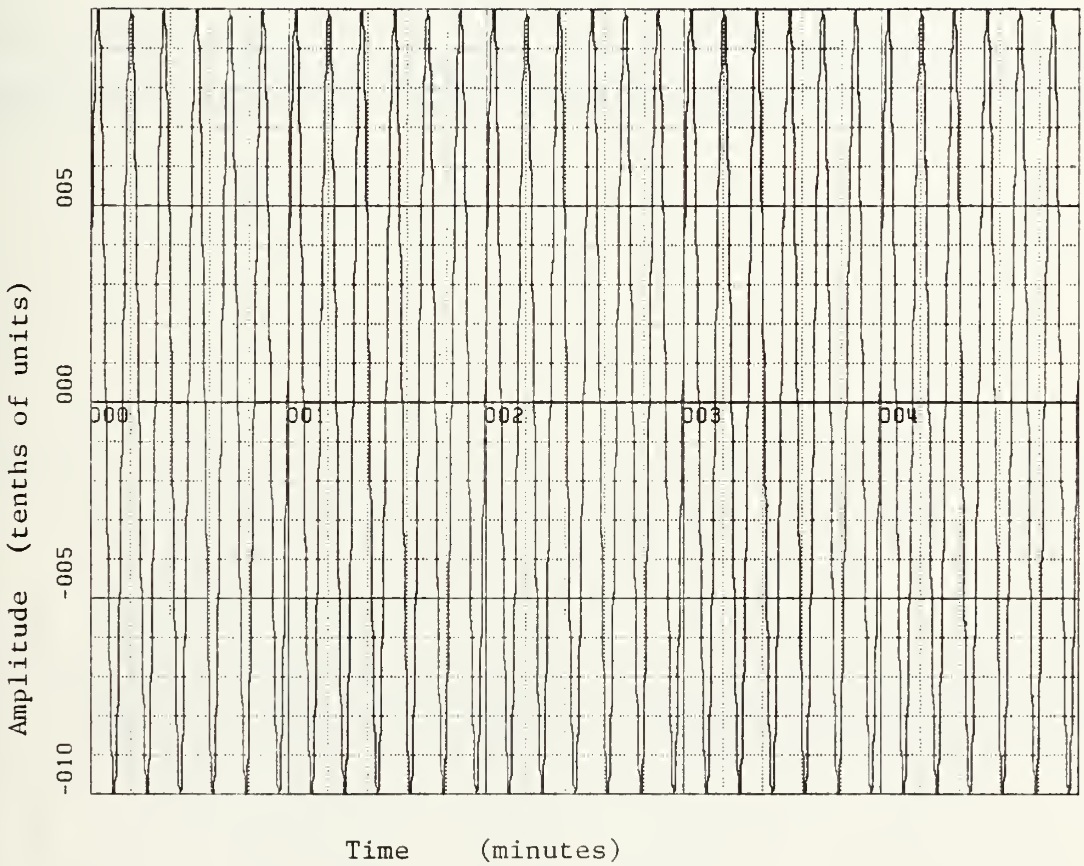

Figure 5.3: Input Signal to Digital Filter Program. A Sinusoid of Frequency 0.1 HZ and Amplitude $\pm$ 1.

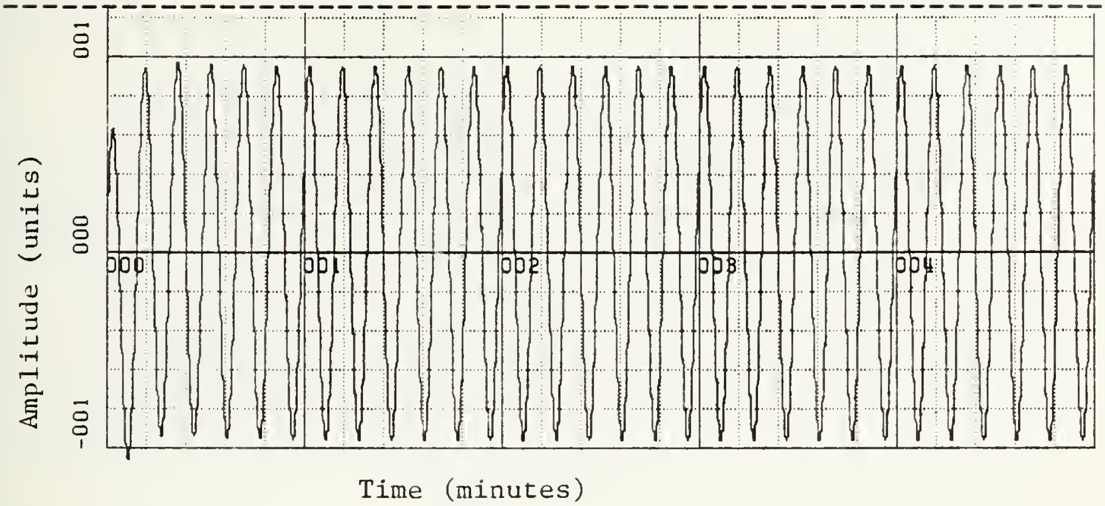

Figure 5.4: Output of First Stage of Digital Filter Program.

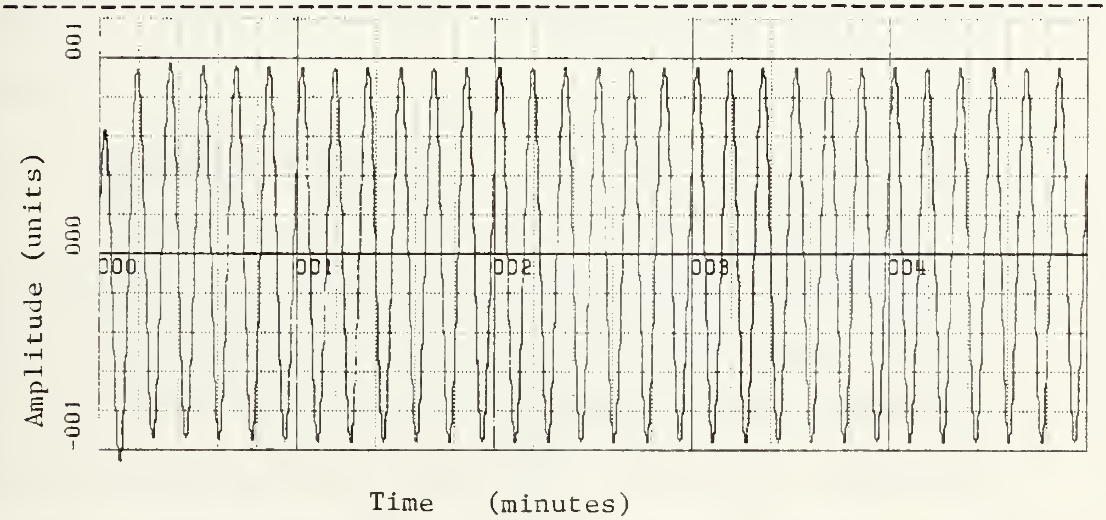

Figure 5.5: Output of Second Stage of Digital Filter Program.

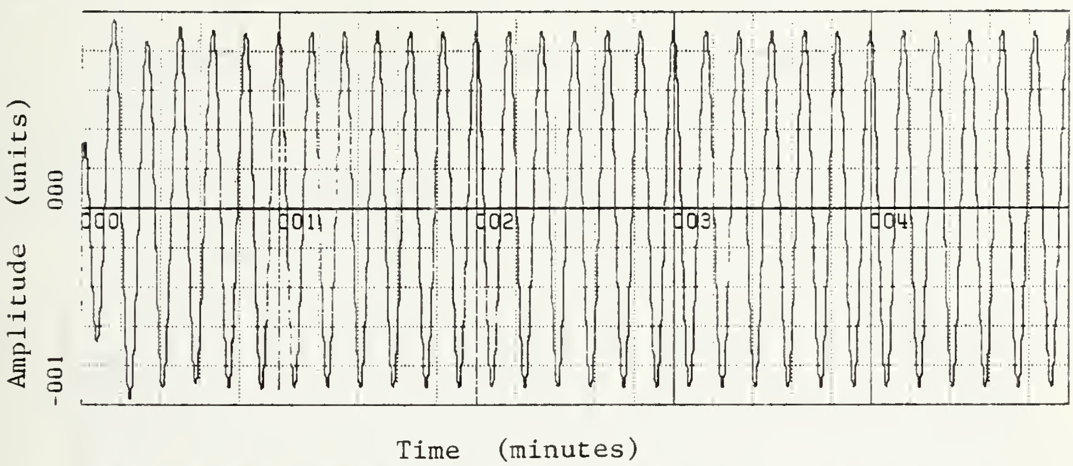

Figure 5.6: Output of Third Stage of Digital Filter Program.

This simulation was run with inputs of sinusoids of various frequencies in order to check the stability of the filter design at frequencies throughout the operating range of the AN/ASQ-81 magnetometer. In all cases, the design was stable, and the expected amplitude changes and phase shifts occured.

2. Noiselike Inputs

The simulation was also run with inputs consisting of a sinusoid of a frequency which should be passed through the AN/ASQ-81 added to sinusoids of frequencies which should have been filtered by the magnetometer and random noise. The filter performed as expected, with the sinusoid of a passable frequency passed by the filter, and spurious noise and sinusoids attenuated severely. The results of a simulation consisting of a sinusoid of passable frequency, a

filterable sinusoid, and uniformly distributed random noise, all of amplitude $\pm$ 1, are presented in Figures 5.7 through 5.10 below.

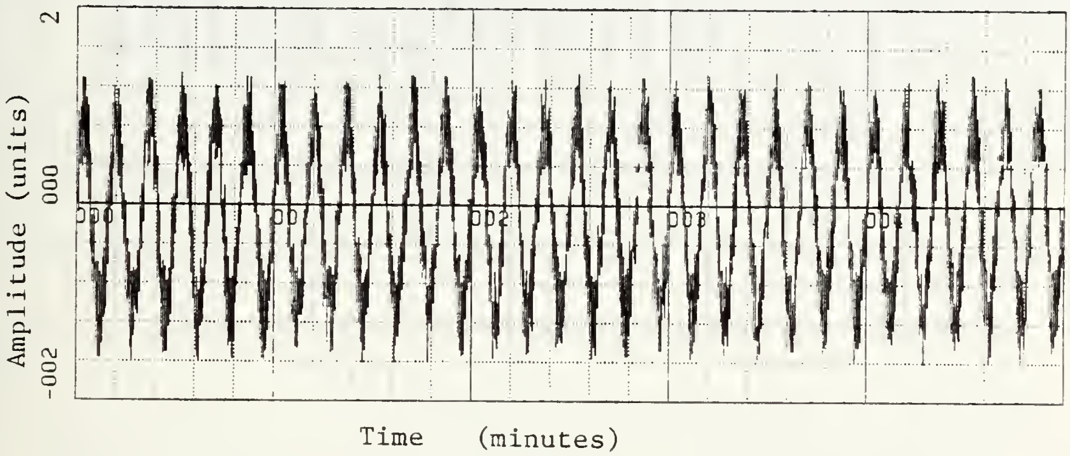

Figure 5.7: Input to Filter - 0.1 HZ Sinusoid, 10 HZ Sinusoid, Uniformly Distributed Random Noise of Amplitude ± 1.

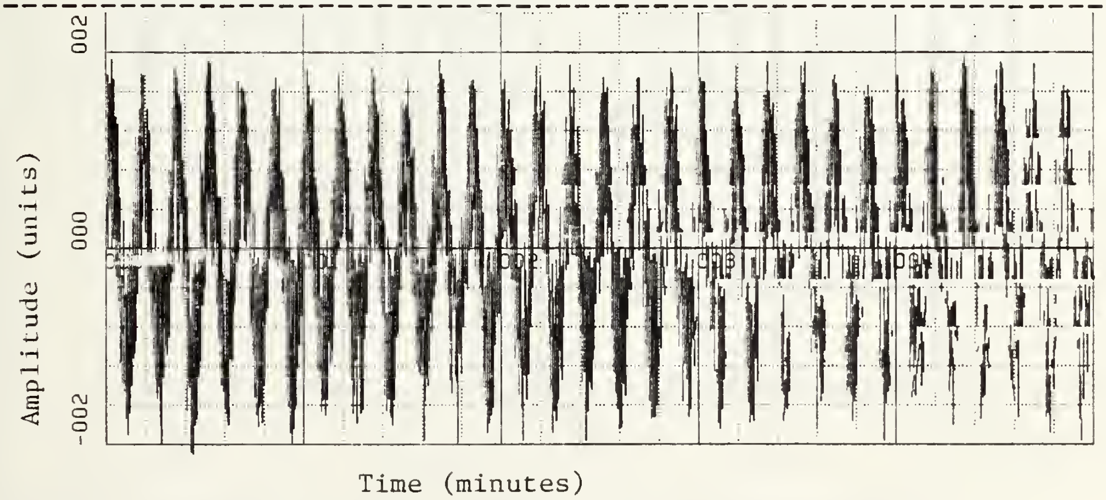

Figure 5.8: Output of First Filter Stage.

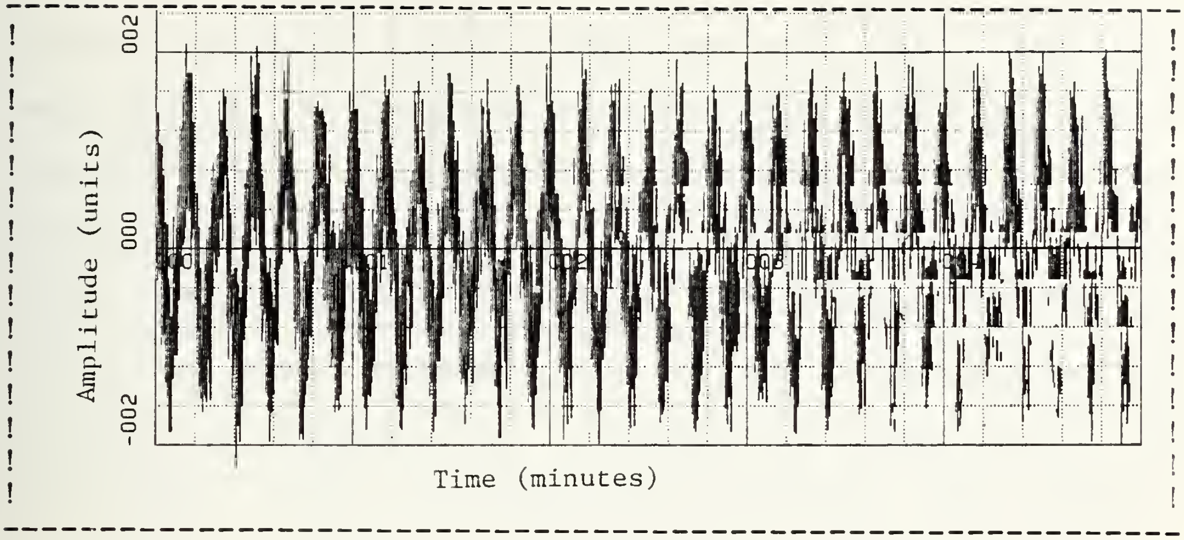

Figure 5.9: Output of Second Filter Stage.

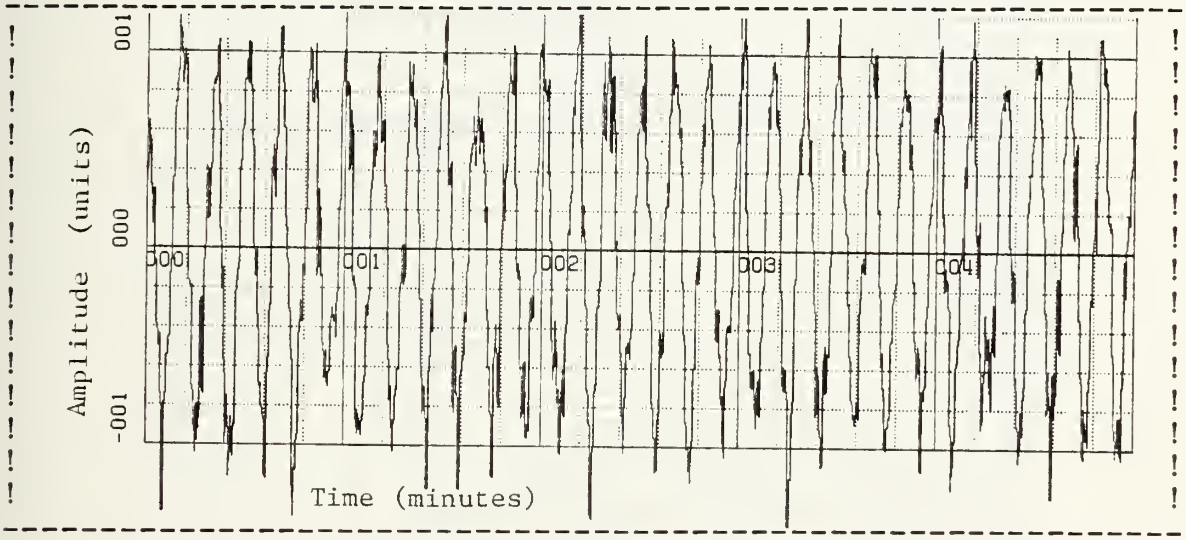

Figure 5.10: Final Filter Stage Output.

As can be seen, the digital filter program succeeds in filtering out random noise and signals of frequency components above the band pass of the magnetometer.

In order to ensure that the digital filter representation of the magnetometer has the same amplitude

versus frequency characteristics of the AN/ASQ-81 magnetometer, a simulation program was written which inputs sinusoids of varying frequencies and computes the Root Mean Square (RMS) value of the filter output and the signal input, then computes the decibel (dB) attenuation of the filter at that frequency. A copy of this program is included in Appendix G. A plot was made of the dB attenuation versus frequency for the filter and compared

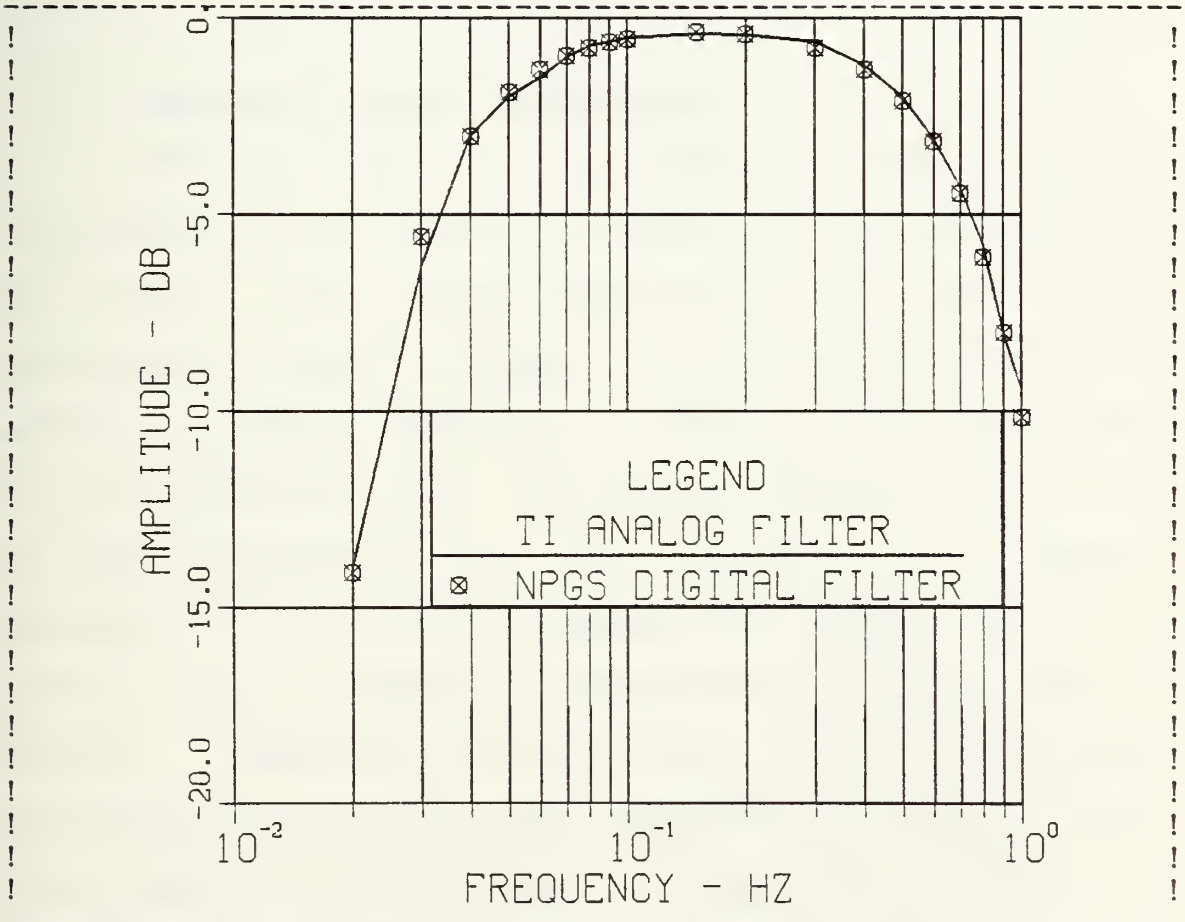

Figure 5.11: Plot of Attenuation Versus Frequency for Sinusoidal Inputs for Digital Filter and Analog Filter.

with the measured frequency performance of the AN/ASQ-81 magnetometer, which was supplied by Texas Instruments, Inc., and does not include the effects of the fixed high pass filter. Consequently, the data shown in Figure 5.11 is a comparison of the data supplied by Texas Instruments and the output of the test program, which also does not include the fixed high pass filter. As can be seen, the performance of the filter is extremely similar to that of the magnetometer itself.

3. Anderson Function Simulations

The next step in the simulation phase was the introduction to the filter of Anderson function simulations. The shape of the signal amplitude of the output of a magnetometer passing through the sphere of influence of a magnetic anomaly (submarine) is a function of the dip angle of the geomagnetic field, the magnetic heading of the track of the magnetometer (or the aircraft), the magnetic heading of the anomaly (submarine) dipole, and the lateral range between the magnetometer (aircraft) and the anomaly. Anderson functions [Ref. 9] are mathematical representations of three basic components of signals which, when taken in various linear combinations, describe the shape of these anomaly signals. The equations for the Anderson functions are:

(First Anderson Function) $f_0 = 1/(1 + B^2)^{5/2}$

where $B = \dfrac{(\text{velocity})\text{x}(\text{time})}{\text{range at CPA}}$,

or, a dimensionless parameter defined as the distance traveled along the magnetometer (aircraft) track divided by the slant range at closestpoint of approach (CPA)

(Second Anderson Function) $f_1 = B \times f_0$

(Third Anderson Function) $f_2 = B \times f_1 = B^2 \times f_0$

The Anderson functions were introduced into the filter program in a noise-free signal environment in order to observe the output signal and ensure that it was a "MAD-like" signal. A rigorous determination of the actual output signal would have been extremely difficult to obtain, so a comparison was made with the output of a computer simulation program provided to NPS by Mr. Joe Rice of Texas Instruments. When the sampling rate of the program was adjusted to equal that of the Texas Instruments program, 8 HZ, the two program outputs were observed to be very similar. The Anderson function simulation inputs and outputs of the program are depicted in Figures 5.12 through 5.18. The Texas Instruments program outputs were obtained in the form of time series plots of discontinuous data points, and were therefore not conducive to replotting for comparison.

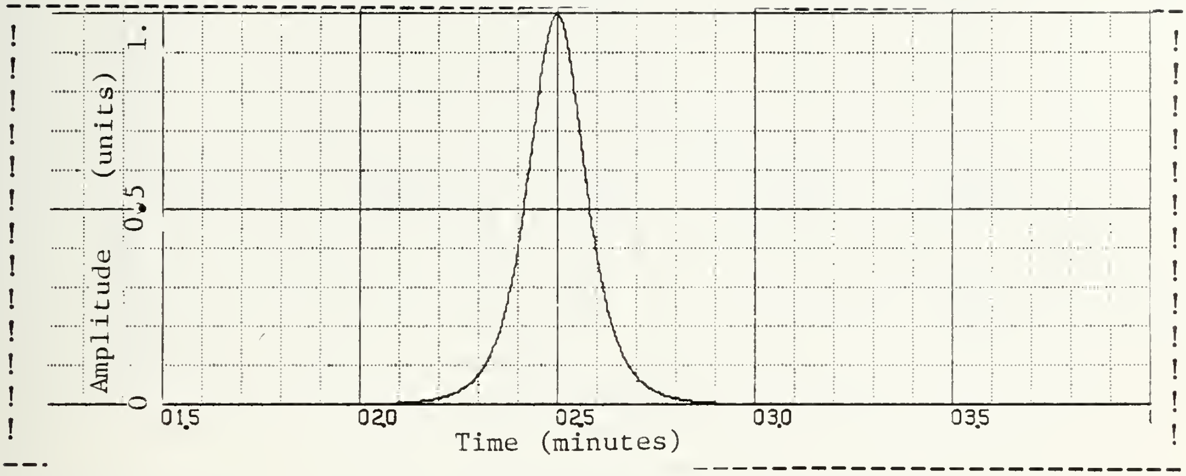

Figure 5.12: First Anderson Function Input. CPA at Time 2.5 Minutes.

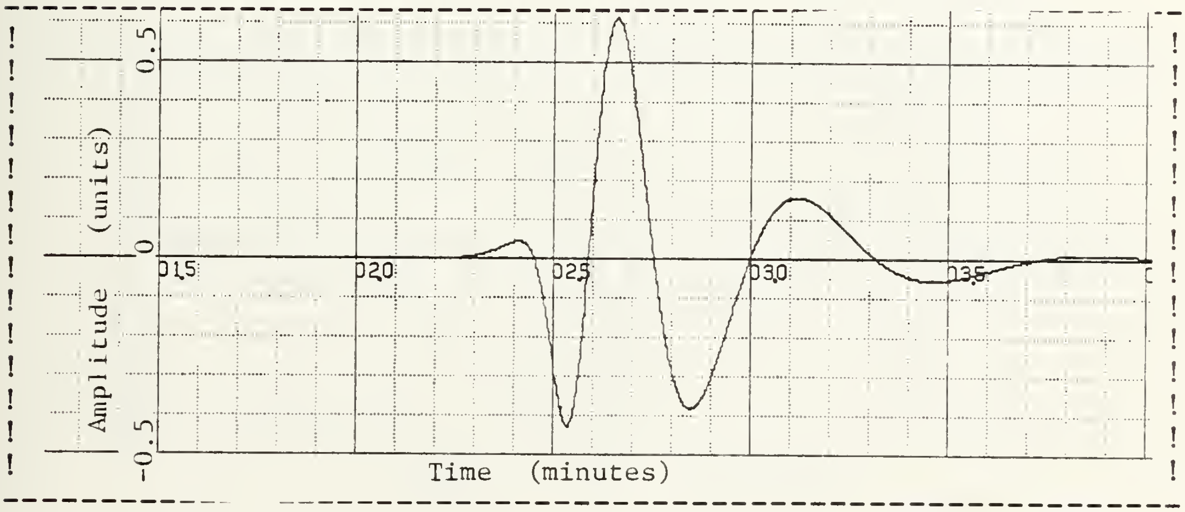

Figure 5.13: Filter Output for First Anderson Function Input.

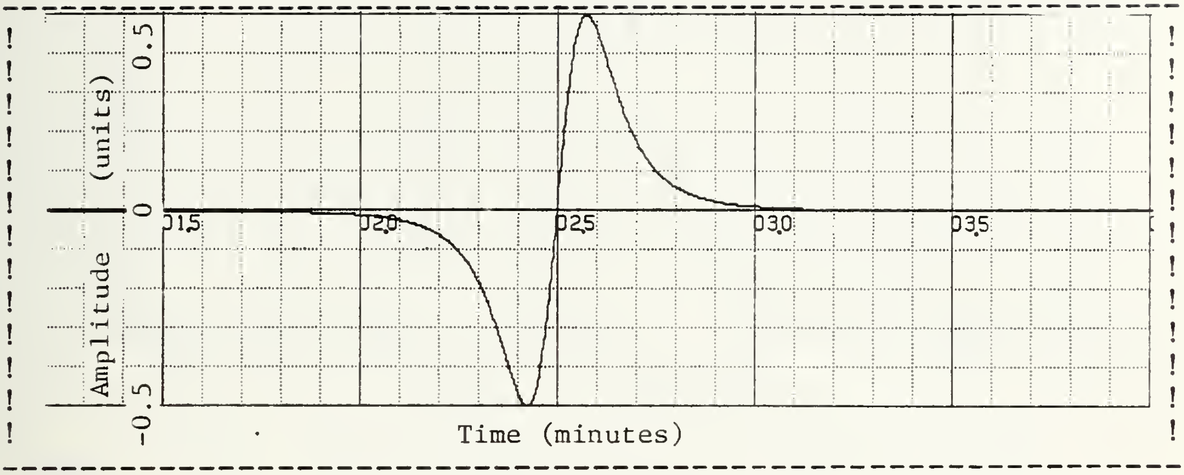

Figure 5.14: Second Anderson Function Input. CPA at Time 2.5 Minutes.

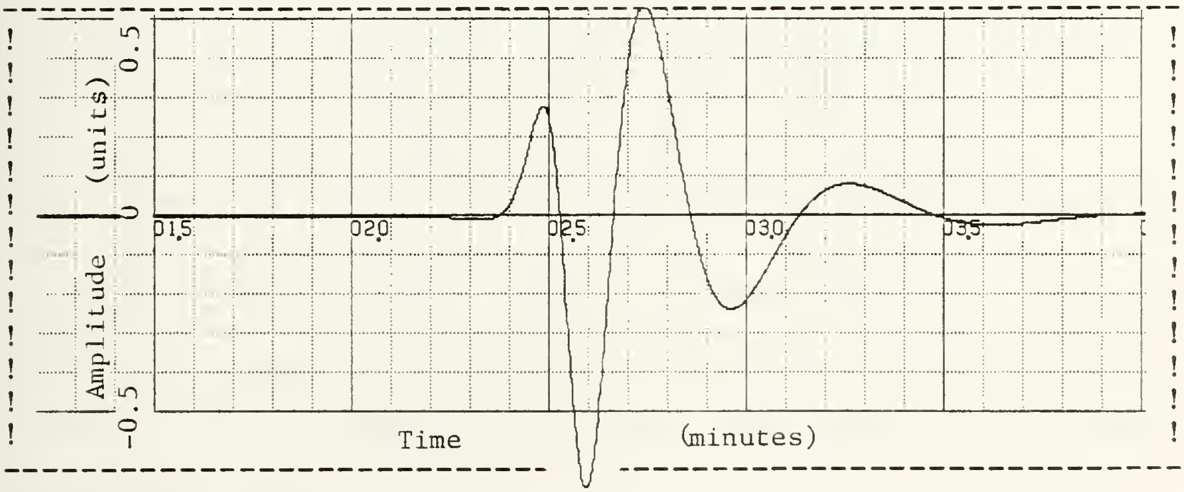

Figure 5.15: Filter Output for Second Anderson Function.

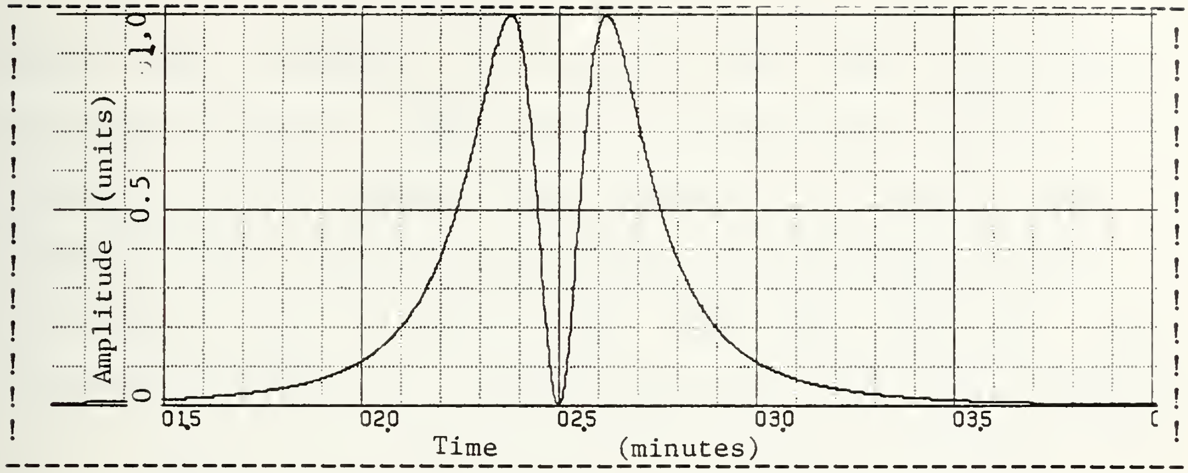

Figure 5.16: Third Anderson Function Input. CPA at Time 2.5 Minutes.

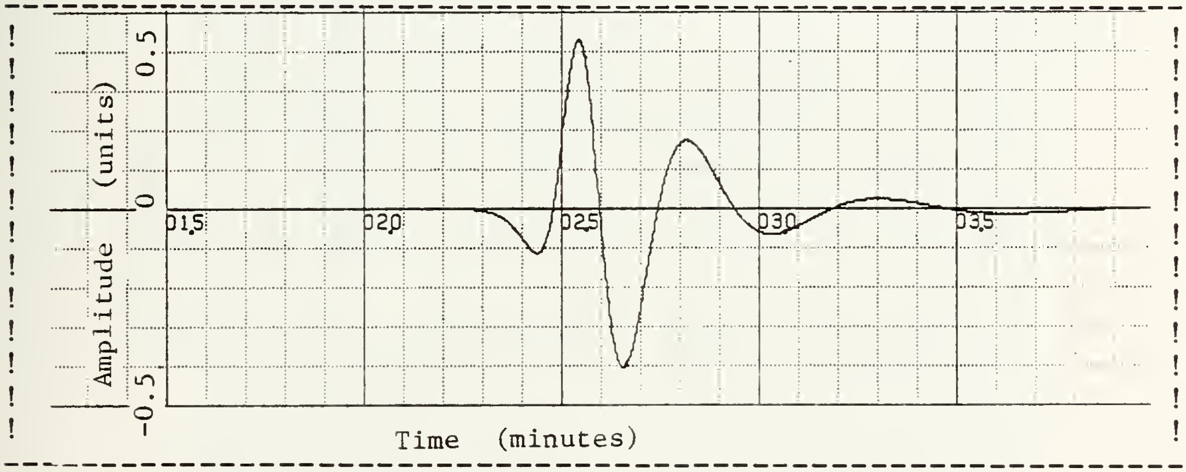

Figure 5.17: Filter Output for Third Anderson Function.

The filter output for all three Anderson function inputs did appear to be "MAD-like" signals, and did closely resemble the simulation output obtained from Texas Instruments, Inc.

4. Impulse Function Response

The response of the filter program was also observed when the input was a unit impulse function. Again, the

output was compared to that of the Texas Instruments' computer program. The outputs of the two programs were observed to be, again, very similar, as can be seen in Figure 5.18, where the response of the NPGS filter is represented by a solid line and that of the Texas Instruments filter by a chain-dash line. The abrupt "jumps"

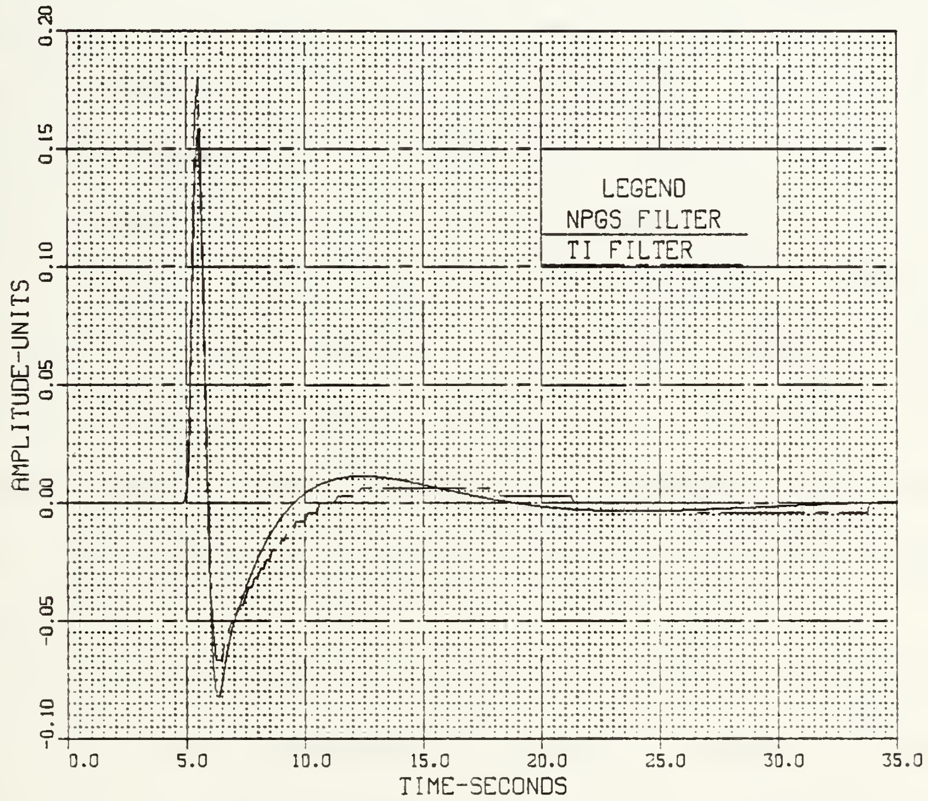

Figure 5.18: Impulse Response of Filters.

of the Texas Instruments response are due to the translation of the output plot supplied to this plot. The plot supplied by Texas Instruments was, again, discontinuous points of poor resolution, and it was necessary to interpolate values

in order to generate Figure 5.18. This resulted in the broken appearance of the plot. Even so, the similarity of the outputs can be observed.

B. EQUIPMENT SETUP

Following the simulation phase of the experiment, actual magnetic field measurements were introduced to the filter in order to test the response of the filter. Magnetic field measurements were made at the La Mesa field test site near the Naval Postgraduate School in Monterey. The output of an AN/ASQ-81 magnetometer, a Schonstedt magnetic field sensor, and the school's coil sensor, oriented along the Earth's magnetic field, were pulse code modulated (PCM) and transmitted via VHF radio to recording devices at the Postgraduate school. The recording of a two hour long data collection period was transferred to digital data tape for use by the school's IBM3033 general purpose mainframe computer.

In the first test of the digital filter program, the output of the Schonstedt sensor, which represents fluctuations of the Earth's total field, was used as the input to the computer program. A comparison of the output of the computer program, with this approximation to the total field fluctuations as input, to the output of the AN/ASQ-81 should provide an indication of the proper functioning of the computer filter program. The results of

the test are shown in Figures 5.19 through 5.21 on the pages following. Figure 5.19, the Schonstedt sensor output, shows several instances of PCM dropouts, that is, occasions where the pulse code modulation signal was not correctly read by the computer for some reason. At such occurances, the data point value used by the computer is a random number and does not reflect the true value of the data. The problem with these PCM dropouts is that the computer does not recognize them as invalid data points and will use them in computations. This can (and does) cause problems in the computation of Fourier transforms, spectral characteristics, etc. Additionally, this will also impact the proper functioning of the digital filter program which is the subject of this thesis. PCM dropouts are visible at times 6, 8, 142, and 220 through 226 seconds on the plot of the Schonstedt sensor output. An examination of Figure 5.20, the filter program output, reveals the programs attempt to "follow" these PCM dropouts. It should be recalled that previous simulations indicated the filter's tendency to "follow" sudden changes in the input signal, with a relaxation time required for the filter to steady out. This effect is apparent in the output of the filter program at times corresponding to those of the PCM dropouts in the Schonstedt sensor's time series plot. It can be seen that this overshoot tendency resulted in an output significantly different from the actual AN/ASQ-81 output at these times.

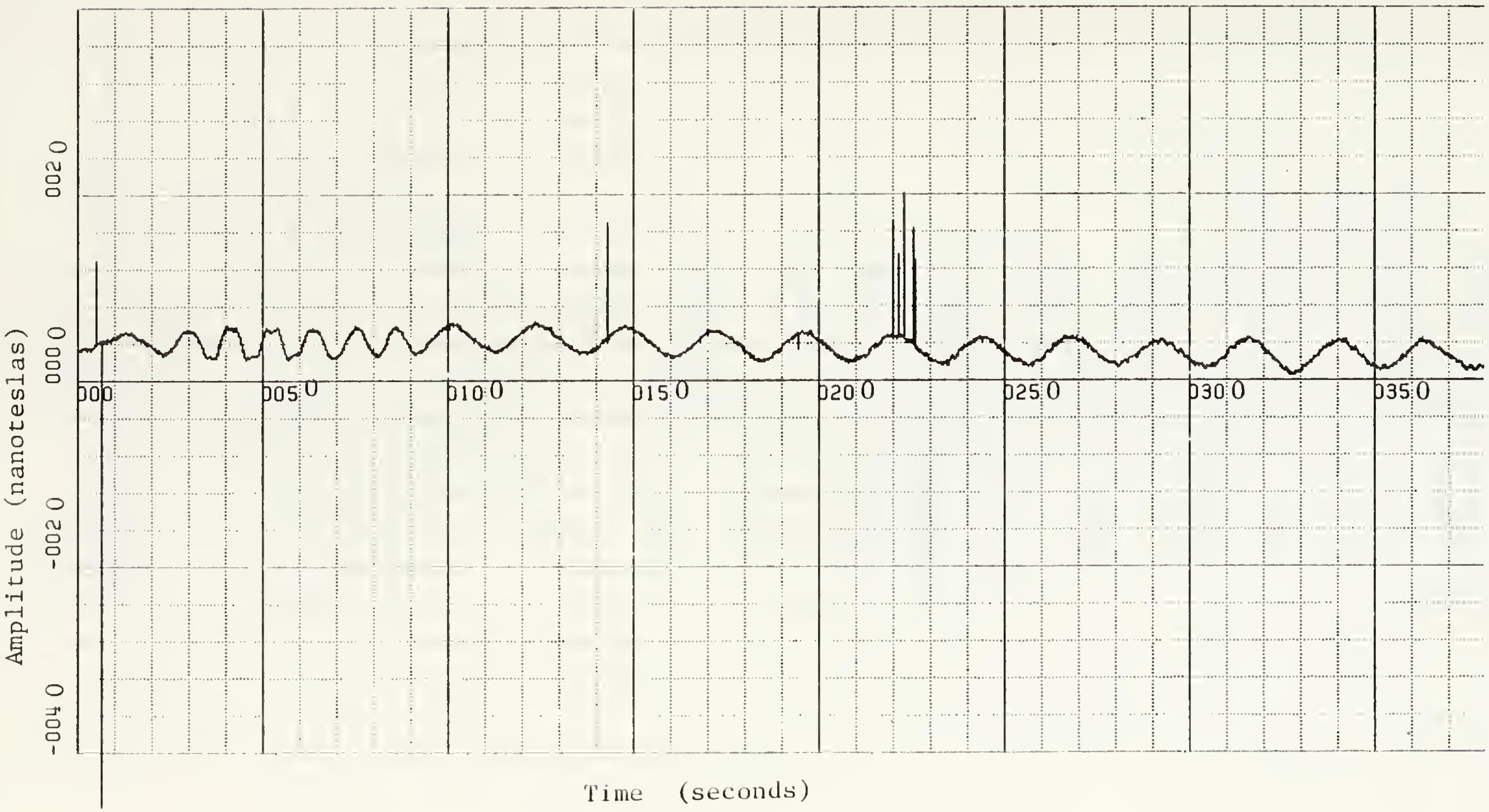

Figure 5.19: Schonstedt Coil Time Series Output

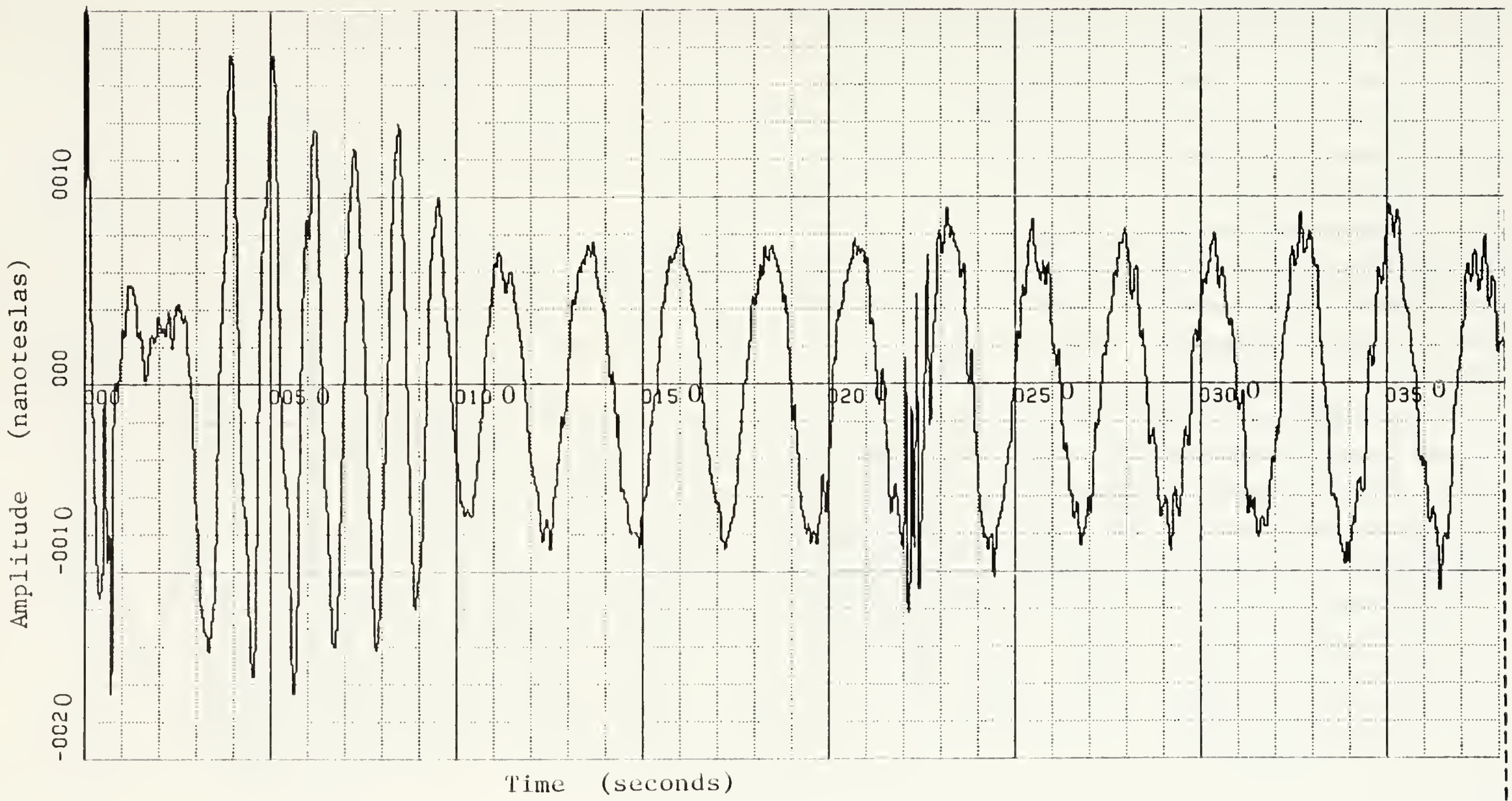

Figure 5.20: Program Time Series Output With Schonstedt Coil as Input

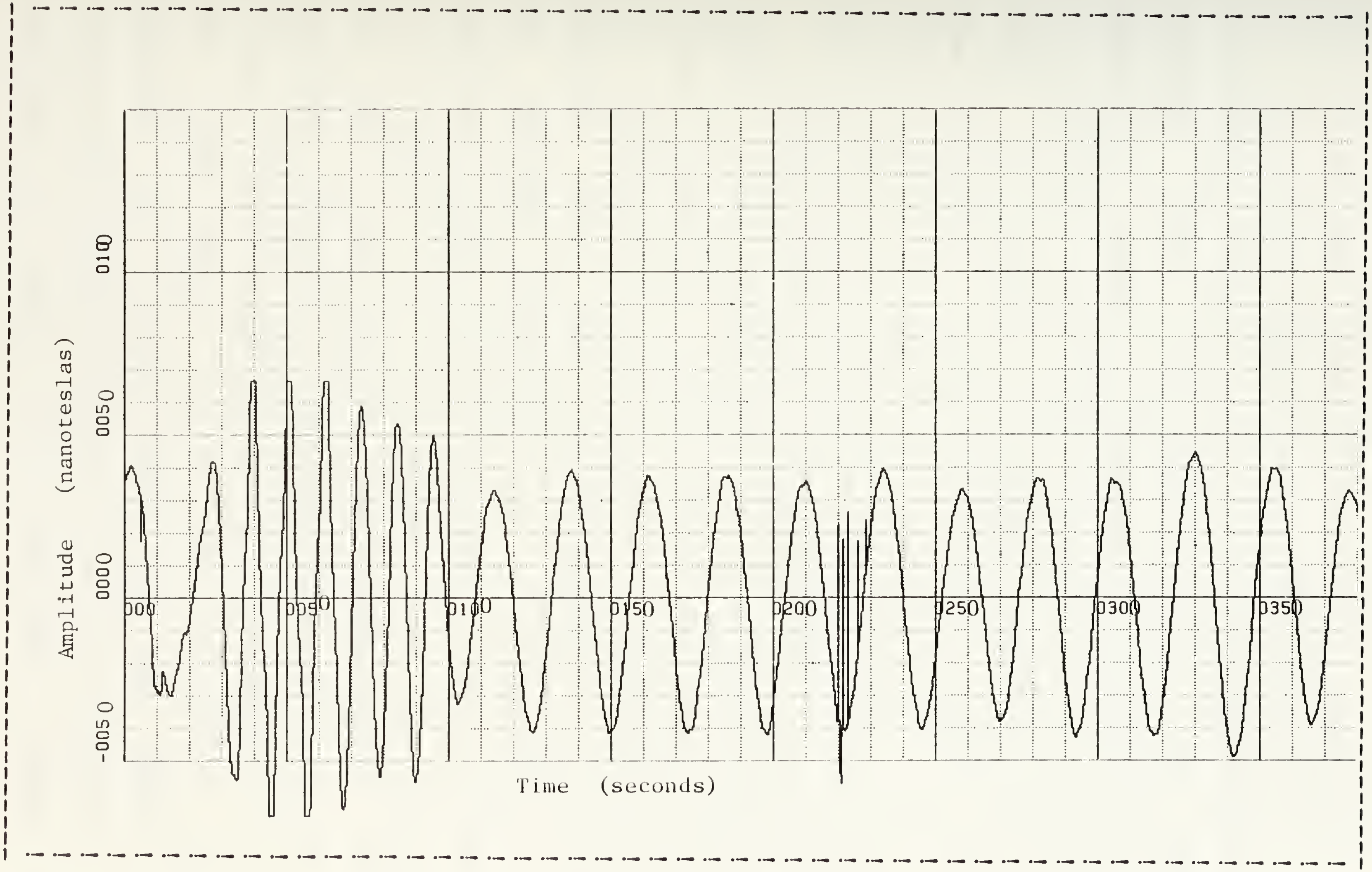

Figure 5.21: ASQ-81 Time Series Output

If the PCM dropout induced differences are neglected, it can be seen that the shape of the output of the filter program is remarkably similar to that of the AN/ASQ-81 magnetometer, although noisier. Note that the output of the AN/ASQ-81 magnetometer exceded the maximum voltage amplitude which the pre-amplifiers of the data collection system were able to handle and resulted in a truncated signal from time 40 to time 60 seconds. It can still be seen, however, that the filter program output is very similar to the time signal which would have been displayed without this truncation.

It should be noted that the amplitudes of the time series signals of the program output and the AN/ASQ-81 magnetometer differ considerably. In the case of the program output, the peak amplitudes are on the order of 1.4 nanoteslas along the vertical scale, while the peak amplitudes of the output of the AN/ASQ-81 magnetometer are on the order of 0.7 nanoteslas along the vertical scale. This is because the input signal to the filter program is an approximation to the total field difference time series signal, and some amplitude difference could reasonably be expected. The intent of this initial test was to investigate the output time series shape, and an exact correlation was not expected. It is worth noting that the digital filter program will perform its function on any time series signal, regardless of units. This means that a signal may be

operated upon either before or after conversion from whatever units it was originally measured to magnetic field strength units.

Therefore it appears that the digital filter program is functioning properly. When a close approximation to the fluctuations of the total field time series signal is used as the input to the computer program, the output of the program is similar to the time series output of an AN/ASQ-81 magnetometer.

The final stage in the testing process was a conversion of the time series output voltage signal of the coil antenna sensor, which was aligned along the Earth's magnetic field, into a total field fluctuation time series representation for the same time period as before, and then to use this as the input to the digital filter program. A comparison of the resultant time series output of the program with the actual AN/ASQ-81 magnetometer output would validate the proper functioning of the program.

Conversion of the time series antenna sensor output voltage signal into total field fluctuations in nanoteslas was accomplished through the use of a computer program designed by Capt. Kurt Stevens, USAF, a student at the Naval Postgraduate School, as his Master's thesis [Ref 10]. The output voltage time series is stored in an array, then a Fourier transform is performed on the stored data, resulting in the Fourier spectrum of the data. This spectrum is

corrected for the characteristics of the coil antenna sensor to obtain the Fourier spectrum of the total field data. A reverse Fourier transform gives the time series signal for total magnetic field in nanoteslas.

This time series signal was used as the input to the digital filter program and compared with the output of the AN/ASQ-81 magnetometer. Figures 5.22 through 5.24 show the raw coil antenna data, the total field time series data, and the program output time series for a 6 minute period of the test. Figure 5.22 shows the raw coil antenna data series. The number of PCM dropouts should be noted, as these will influence the performance of the filter program. Figure 5.23 shows the computed total field time series. Note that the PCM dropouts evident on the raw time series plot are evident on the computed total field time series plot also, and thus inputted to the filter program as valid data points. Additionally, there are two "jumps" in the plot of total field fluctuation (Figure 5.23) which are also inputted to the filter program as valid data points. These "jumps" are located at 128 and 256 seconds and are caused by the method of processing blocks of data for the conversion to total field fluctuation. A block of 128 seconds of data is processed at a time, and the results of each block are stored in an array. This results in a slight amplitude difference between the last data point of one block and the

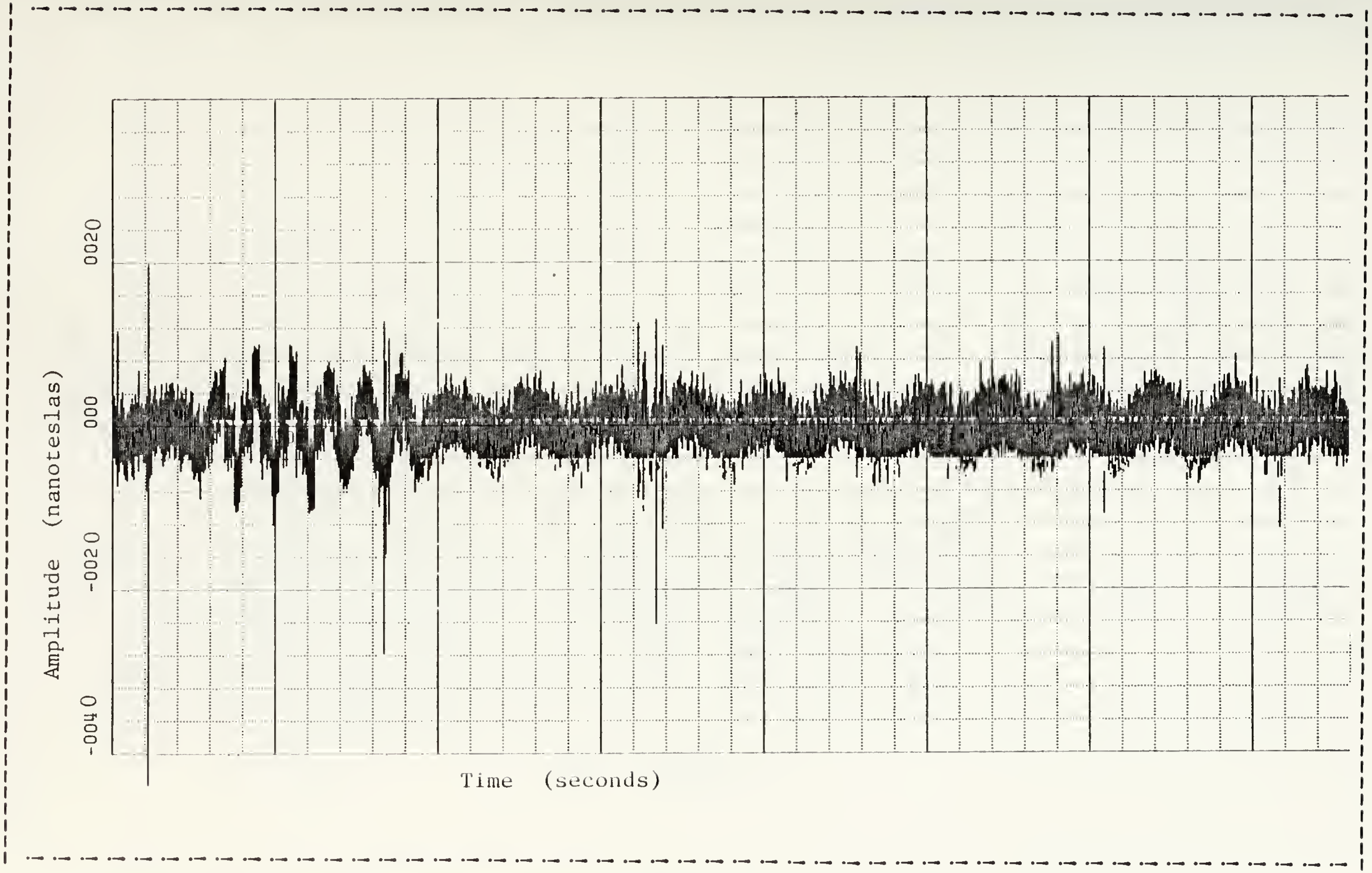

Figure 5.22: Raw Coil Antenna Time Series Output

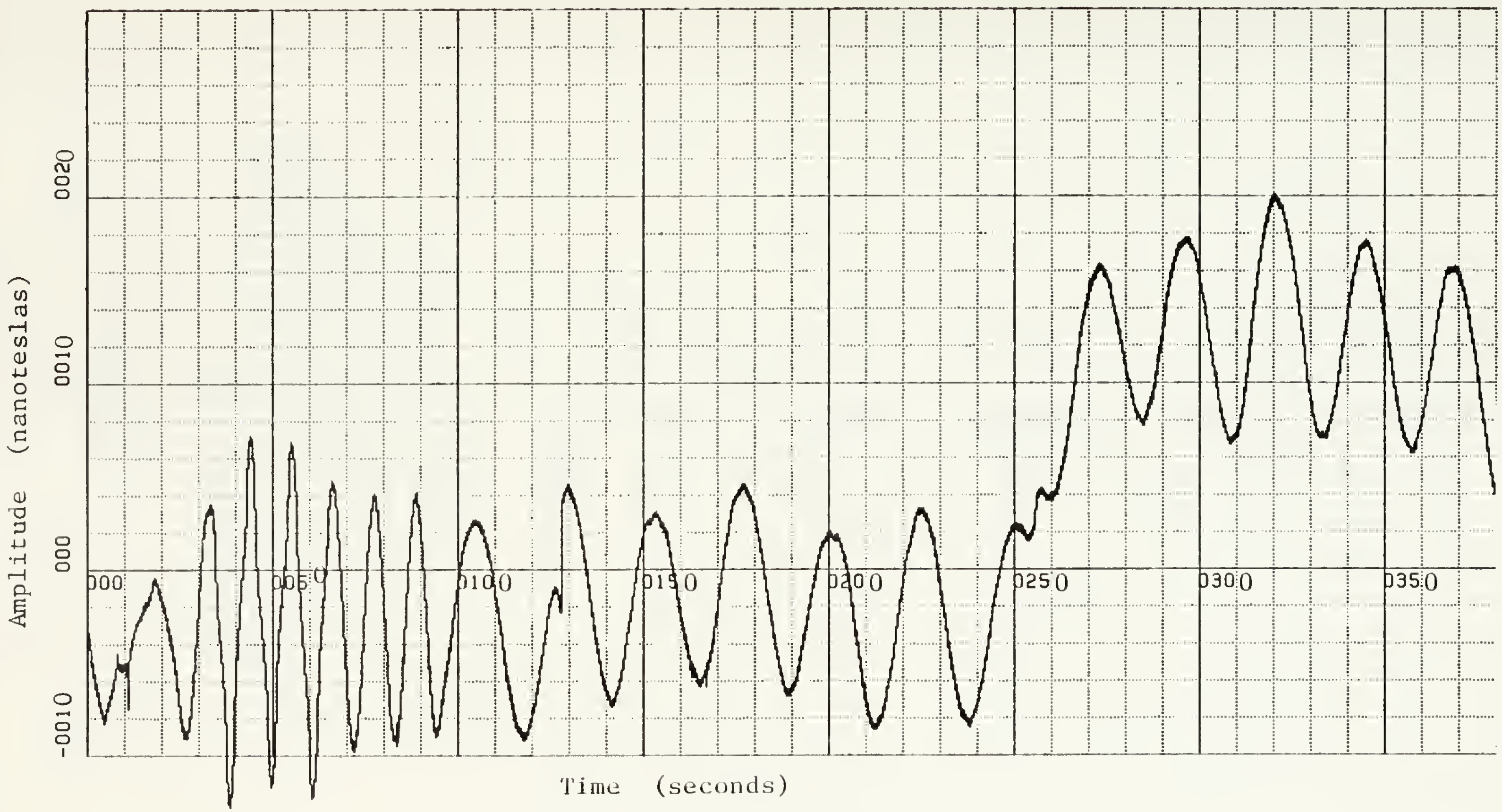

Figure 5.23: Coil Antenna Difference Field Time Series Output

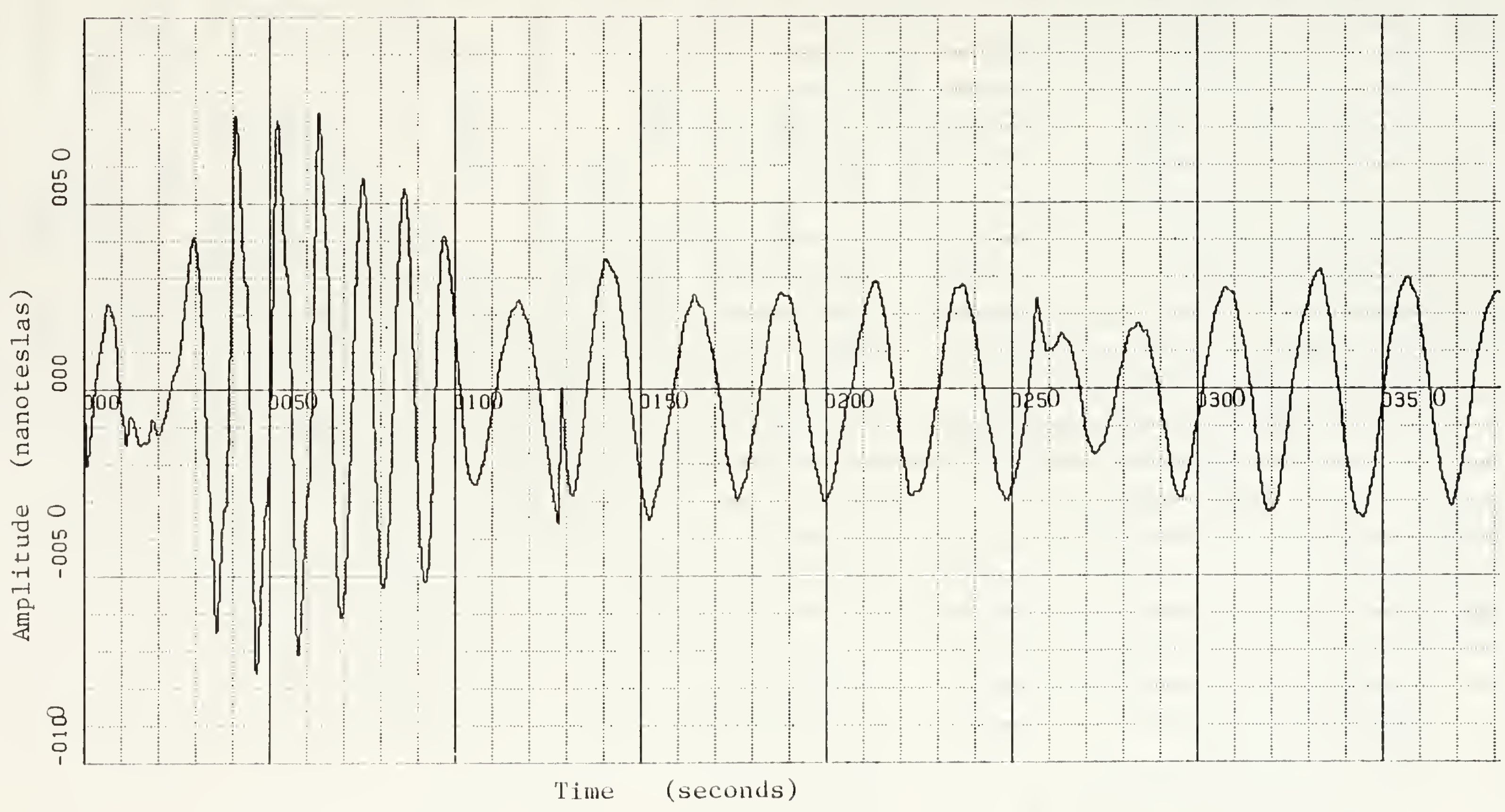

Figure 5.24: Program Time Series Output With Coil Antenna Difference Field Time Series as Input.

first data point of the next block of data. This slight difference is manifested as a signal jump.

A comparison of Figures 5.24 (program output) and 5.21 (AN/ASQ-81 output) show that the filter program gives a time series output very similar to that of the actual magnetometer. The first 20 seconds of the program output is somewhat dissimilar to that of the AN/ASQ-81, due either to the initial "start up" delay of the filter program or to distortion of the total field fluctuation time series. There is a PCM dropout at time 11 seconds which contributed to the distortion.

Following this, however, it can be seen that the program output is very similar to that of the magnetometer, except at 128 and 256 seconds, which show the effects of the false signal jumps caused by the total field fluctuation conversion. There is also a noticeable four to five second time delay between the AN/ASQ-81 output and that of the filter program. As this delay is not evident in a comparison of the AN/ASQ-81 output and that of the filter program with the Schonstedt sensor as the input, it can be inferred that this time delay is caused either by the program which converts the raw coil data to total field fluctuation data, or by a phase (and hence time) change of the voltage signal due to the coil sensor itself. A comparison of the raw coil data in Figure 5.22 to the converted coil data in Figure 5.23 indicates no time shift,

and hence the deduction can be made that there is a time delay inherent within the coil sensor itself.

Other than the differences of the four to five second time delay and the distortions caused by the false signal jumps, the program output is extremely similar to the output of the AN/ASQ-81 magnetometer.

VI. CONCLUSIONS

The intent of this thesis was to design and test a digital filter computer program which would, when given a time series input of fluctuations in the total magnetic field, deliver an output time series representation of the output of an AN/ASQ-81 magnetometer. This purpose has been realized.

The computer program contained in Appendix I has been proven to output a time series signal which is very similar to that of the magnetometer. The major limitations of the output signal are a finite time delay of about five seconds between the AN/ASQ-81 magnetometer signal and the output signal of the program, a sensitivity of the program to false data points such as those caused by PCM dropouts and false signal jumps caused by processing large data blocks, and the inherent limitations of the program caused by its dependence on the use of digital data tapes and the IBM 3033 mainframe computer.

The five second time delay is not considered to be an important limitation to the program, as it was intended as a research tool for programs currently in progress at the Naval Postgraduate School. Instances where this time delay might become important would be in areas of simultaneous comparison of the program output signal with an actual magnetometer, in target location algorithms using time

delays, or in correlation studies between different sensors. In correlation studies using coil sensors, the effects of the time delay would cancel out, as all coil outputs would be similarly delayed. In target location algorithms, the target location errors due to the time delay could be adjusted for simply while in computer simulation, and flight testing could not reasonably be accomplished without the use of an actual magnetometer as the sensor. Lastly, in a comparison of the program output with an actual sensor, the time delay can, again, be compensated for. In short, these limitations are not considered excessive, especially as the apparent cause for the delay is not the filter program.

In the primarily intended purpose of the filter program, magnetic noise studies, the time delay is not considered to be a problem.

The problem of false data points caused by PCM dropouts and signal jumps due to conversion to total field fluctuations is more serious. False data points cause inaccuracies in the output time series and could adversely affect later projects. Unfortunately the PCM dropout problem is one which is endemic to the data collection system presently being used at the postgraduate school, and not to the filter program itself. It is imperative that users of this program are aware of the PCM dropout problem and of the effects it may entail upon their specific

research. A large number of PCM dropouts in a time series could render that series unusable. Similarly the case of the false data jumps caused by conversion to total field fluctuations is not within the filter program. Further investigation of this problem is necessary in order to eliminate it.

The last problem, that of reliance upon the digital data tape/IBM 3033 computer system, is, like the PCM problem, one which is not endimic to the filter program but rather to the data collection system being used. A change of data collection system may, at some future time, remove the reliance upon the PCM/digital tape/IBM 3033 data system (and hence too the data block conversion problem which results in falso data jumps), but this is unlikely at this time. Users should be aware of this dependence and of possible effects upon specific research projects.

APPENDIX A

AN/ASQ-81 FILTER TRANSFER FUNCTIONS

Fixed High Pass Transfer Function:

$$H(S) = \frac{80\ S^2}{80\ S^2 + 20\ S + 1}$$

Selectable High Pass Transfer Functions:

A. 0.04 HZ H(S) =

$$\frac{40.82834\ S^2}{40.82834\ S^2 + 12.52096\ S + 1} \times \frac{45.28317\ S^2}{45.28317\ S^2 + 11.00999\ S + 1}$$

$$\times \frac{57.576688\ S^2}{57.576688\ S^2 + 7.41498\ S + 1}$$

B 0.06 HZ H(S) =

$$\frac{18.14591\ S^2}{18.14591\ S^2 + 8.34727\ S + 1} \times \frac{20.12587\ S^2}{20.12587\ S^2 + 7.33999\ S + 1}$$

$$\times \frac{25.58964\ S^2}{25.58964\ S^2 + 4.94332\ S + 1}$$

C. 0.08 HZ H(S) =

$$\frac{10.20708\ S^2}{10.20708\ S^2 + 6.26045\ S + 1} \times \frac{11.32080\ S^2}{11.32080\ S^2 + 5.50500\ S + 1}$$

$$\times \frac{14.39417\ S^2}{14.39417\ S^2 + 3.70749\ S + 1}$$

D. <u>0.10</u> <u>HZ</u> H(S) =

$$\frac{6.53253\ S^2}{6.53253\ S^2 + 5.00836\ S + 1} \times \frac{7.24531\ S^2}{7.24531\ S^2 + 4.40400\ S + 1}$$

$$\times \frac{9.21227\ S^2}{9.21227\ S^2 + 2.96599\ S + 1}$$

Selectable Low Pass Transfer Functions

A. <u>0.2</u> <u>HZ</u> H(S) =

$$\frac{1}{0.3143\ S^2 + 1.0741\ S + 1} \times \frac{1}{0.2501\ S^2 + 0.6209\ S + 1}$$

B. <u>0.4</u> <u>HZ</u> H(S) =

$$\frac{1}{0.07858\ S^2 + 0.53706\ S + 1} \times \frac{1}{0.06252\ S^2 + 0.31044\ S + 1}$$

C. <u>0.6</u> <u>HZ</u> H(S) =

$$\frac{1}{0.03492\ S^2 + 0.35804\ S + 1} \times \frac{1}{0.02779\ S^2 + 0.20696\ S + 1}$$

APPENDIX B

AN/ASQ-81 Z TRANSFORM FILTER TRANSFER FUNCTIONS FOR DIRECT FORM I REALIZATION

For fixed high pass filter:

$$H(Z) = \frac{BFHP0 + BFHP1*Z^{-1} + BFHP2*Z^{-2}}{1 - AFHP1*Z^{-1} - AFHP2*Z^{-2}}$$

where BFHP0, BFHP1, BFHP2, AFHP1, AFHP2 are constants tabulated in Appendix D.

For selectable high pass filter:

$$H(Z) = \frac{BSHP0 + BSHP1*Z^{-1} + BSHP2*Z^{-2} + BSHP3*Z^{-3} + BSHP4*Z^{-4} + BSHP5*Z^{-5} + BSHP6*Z^{-6}}{1 - ASHP1*Z^{-1} - ASHP2*Z^{-2} - ASHP3*Z^{-3} - ASHP4*Z^{-4} - ASHP5*Z^{-5} - ASHP6*Z^{-6}}$$

where, for low frequency cutoff of 0.04 HZ:

BSHP0 = 0.99471378

BSHP1 = -5.9682827 ASHP1 = 5.9894021

BSHP2 = 14.920707 ASHP2 = -14.947051

BSHP3 = -19.894276 ASHP3 = 19.894225

BSHP4 = 14.920707 ASHP4 = -14.894327

BSHP5 = -5.9682817 ASHP5 = 5.9472141

BSHP6 = 0.99471372 ASHP6 = -0.98945296

For low frequency cutoff of 0.06 HZ:

BSHP0 = 0.9920813	
BSHP1 = -5.9524928	ASHP1 = 5.9841070
BSHP2 = 14.881232	ASHP2 = -14.920650
BSHP3 = -19.841643	ASHP3 = 19.841528
BSHP4 = 14.881232	ASHP4 = -14.841757
BSHP5 = -5.9524927	ASHP5 = 5.9209919
BSHP6 = 0.99208212	ASHP6 = -0.98422128

For low frequency cutoff of 0.08 HZ:

BSHP0 = 0.98945806	
BSHP1 = -5.9367483	ASHP1 = 5.9788144
BSHP2 = 14.841871	ASHP2 = -14.894276
BSHP3 = -19.789161	ASHP3 = 19.788959
BSHP4 = 14.841871	ASHP4 = -14.789365
BSHP5 = -5.9367476	ASHP5 = 5.8948841
BSHP6 = 0.98945802	ASHP6 = -0.97901720

For low frequency cutoff of 0.1 HZ:

BSHP0 = 0.98684156	
BSHP1 = -5.9210493	ASHP1 = 5.9735244
BSHP2 = 14.802623	ASHP2 = -14.867941
BSHP3 = -19.736831	ASHP3 = 19.736516
BSHP4 = 14.802623	ASHP4 = -14.737149
BSHP5 = -5.9210491	ASHP5 = 5.8688889
BSHP6 = 0.9868415	ASHP6 = -0.97384065

For selectable low pass filter:

$$H(Z) = \frac{BSLP0 + BSLP1*Z^{-1} + BSLP2*Z^{-2} + BSLP3*Z^{-3} + BSLP4*Z^{-4}}{1 - ASLP1*Z^{-1} - ASLP2*Z^{-2} - ASLP3*Z^{-3} - ASLP4*Z^{-4}}$$

where BSLP0, BSLP1, BSLP2, BSLP3, BSLP4, ASLP1, ASLP2, ASLP3, ASLP4 are constants tabulated in Appendix D.

APPENDIX C

AN/ASQ-81 Z TRANSFORM FILTER TRANSFER FUNCTIONS DIRECT FORM II REALIZATION

For fixed high pass filter:

$$H(Z) = \frac{BFHP0 + BFHP1*Z^{-1} + BFHP2*Z^{-2}}{1 - AFHP1*Z^{-1} - AFHP2*Z^{-2}}$$

where BFHP0, BFHP1, BFHP2, AFHP1, AFHP2 are constants tabulated in Appendix D.

For selectable high pass filter:

$$H(Z) = ASHP1 \times \frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP3*Z^{-1} - ASHP2*Z^{-2}} \times$$

$$\frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP4*Z^{-1} - ASHP5*Z^{-2}} \times \frac{1 - 2*Z^{-1} + Z^{-2}}{1 - ASHP6*Z^{-1} - ASHP7*Z^{-2}}$$

where ASHP1, ASHP2, ASHP3, ASHP4, ASHP5, ASHP6, ASHP7 are constants and tabulated in Appendix D.

For selectable low pass filter:

$$H(Z) = \frac{BSLP0 + BSLP1*Z^{-1} + BSLP2*Z^{-2} + BSLP3*Z^{-3} + BSLP4*Z^{-4}}{1 - ASLP1*Z^{-1} - ASLP2*Z^{-2} - ASLP3*Z^{-3} - ASLP4*Z^{-4}}$$

where BSLP0, BSLP1, BSLP2, BSLP3, BSLP4, ASLP1, ASLP2, ASLP3, ASLP4 are constants tabulated in Appendix D.

APPENDIX D

Z TRANSFORM REALIZATION DIFFERENCE EQUATIONS

With reference to Figures 3.2 and 4.2, the following difference equations are used to model the AN/ASQ-81 magnetometer filter transfer functions. The input to the fixed high pass filter is called SIG(I), where I is the current data sample. The output of the fixed high pass filter, which is the input to the selectable high pass filter, is YO(I), and the output of the selectable high pass filter, the input to the selectable low pass filter, is called YPO(I). The output of the filter is called ASQ(I). (I-1) denotes a time delay of one sample, and so forth, and the symbol * denotes multiplication.

<u>For the fixed high pass filter:</u>

```
YO(I)=BFHPO*SIG(I) + BFHP1*SIG(I-1) + BFHP2*SIG(I-2) +
          AFHP1*YO(I-1) + AFHP2*YO(I-2)
```

where:

```
BFHPO = 0.9980499222938581

BFHP1 = -1.9960998445877161   AFHP1 =  1.9960983216843922

BFHP2 = 0.998049922238581     AFHP2 = -0.9961013674910398
```

<u>For the selectable high pass filters:</u>

```
XI(I) = ASHP1*YO(I) + ASHP2*XI(I-2) + ASHP3*XI(I-3)

XII(I) = XI(I) + XI(I-2) - 2*XI(I-1)

XIII(I) = XII(I) + ASHP4*XIII(I-1) + ASHP5*XIII(I-2)

XIV(I) = XIII(I) - 2*XIII(I-1) + XIII(I-2)
```

XV(I) = XIV(I) + ASHP6*XV(I-1) + ASHP7*XV(I-2)

YPO(I) = XV(I) - 2*XV(I-1) + XV(I-2)

For the low frequency cutoff at 0.04 HZ:

ASHP1 = 0.994713789347288 ASHP2 = -0.9952196910157882

ASHP3 = 1.9952137256322473 ASHP4 = 1.9962028201103847

ASHP5 = -0.9962082013015601 ASHP6 = 1.9979855321466768

ASHP7 = -0.9979897681491607

For the low frequency cutoff at 0.06 HZ:

ASHP1 = 0.9920821277199399 ASHP2 = -0.9928381306174365

ASHP3 = 1.9928247245317958 ASHP4 = 1.9943056083414792

ASHP5 = -0.9943177045263504 ASHP6 = 1.9969766455640039

ASHP7 = -0.9969861717656565

For the low frequency cutoff at 0.08 HZ:

ASHP1 = 0.9894580558875787 ASHP2 = -0.9904622611337722

ASHP3 = 1.9904384565912725 ASHP4 = 1.9924092959984783

ASHP5 = -0.9924307799269113 ASHP6 = 1.9959666590811389

ASHP7 = -0.9959835860201267

For the low frequency cutoff at 0.10 HZ:

ASHP1 = 0.9869415560096681 ASHP2 = -0.9880929619779491

ASHP3 = 1.9880549117849344 ASHP4 = 1.9905139074397893

ASHP5 = -0.9905474442556225 ASHP6 = 1.9949555824611562

ASHP7 = -0.9949820174650785

For the selectable low pass filters:

ASQ(I) = ASLP1*ASQ(I-1) + ASLP2*ASQ(I-2) + ASLP3*ASQ(I-3)
+ ASLP4*ASQ(I-4) + BSLP0*YPO(I) + BSLP1*YPO(I-1)
+ BSLP2*YPO(I-2) + BSLP3*YPO(I-3) + BSLP4*YPO(I-4)

For the high frequency cutoff at 0.2 HZ:

BSLP0 = 0.0000000452616229

BSLP1 = 0.0000001810464917 ASLP1 = 3.9082436339591027

BSLP2 = 0.0000002715697375 ASLP2 = -5.7285022156249328

BSLP3 = 0.0000001810464917 ASLP3 = 3.7321935213310065

BSLP4 = 0.0000000452616229 ASLP4 = -0.9119356638511430

For the high frequency cutoff at 0.4 HZ:

BSLP0 = 0.0000006918001209

BSLP1 = 0.0000027672004837 ASLP1 = 3.8173771378993420

BSLP2 = 0.0000041508007256 ASLP2 = -5.4670046844062743

BSLP3 = 0.0000027672004837 ASLP3 = 3.4812554457127576

BSLP4 = 0.0000006918001209 ASLP4 = -0.8316389680077603

For the high frequency cutoff at 0.6 HZ:

BSLP0 = 0.0000033463317975

BSLP1 = 0.0000133853271900 ASLP1 = 3.7274299052305002

BSLP2 = 0.0000200779907850 ASLP2 = -5.2152772583906819

BSLP3 = 0.0000133853271900 ASLP3 = 3.2462216641520829

BSLP4 = 0.0000033463317975 ASLP4 = -0.7584278523006615

APPENDIX E

DIGITAL SOFTWARE FOR SIMULATION - DIRECT FORM (SINUSOIDS AS INPUT)

```
//HUETE JOB (1457,1106),' ',CLASS=B
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C                          MGN3
C       THIS PROGRAM IS DESIGNED TO TEST THE ACTION OF THE PRELIMINARY
C       DIGITAL FILTER PROGRAM FOR THE ASQ-81 BY INTRODUCING A SINUSOID
C       AS THE INPUT SIGNAL TO THE FILTER
C
C       SET UP ARRAYS.  SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
C       OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
C       PROGRAM
C
        DIMENSION SIG(3000),ASQ(3000),TRU(3000),TIME(3000)
        DIMENSICN YO(3000),YPO(3000)
        REAL*8 DSEED
        REAL*8 T,AFHP1,AFHP2,BFHP0,BFHP1,BFHP2,A,B,C,D,E,F
        REAL*8 A1,B1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1
        REAL*8 AA,BB,CC,DD,EE,FF,AA1,BB1,CC1,DD1,EE1,FF1,GG1,HH1,II1
        REAL*8 JJ1,KK1,LL1
        REAL*8 ASHP41,ASHP42,ASHP43,ASHP44,ASHP45,ASHP46
        REAL*8 BSHP40,BSHP41,BSHP42,BSHP43,BSHP44,BSHP45,BSHP46
        REAL*8 ASLP61,ASLP62,ASLP63,ASLP64
        REAL*8 BSLP60,BSLP61,BSLP62,BSLP63,BSLP64
        REAL*8 TITLA(12)/'HUETE   ','Y OUT SI','GNAL    ','1457P   ',
       $8*'        '/
        REAL*8 TITLB(12)/'HUETE   ','OUTPUT S','IGNAL   ','1457P   ',
       $8*'        '/
        REAL*8 TITLC(12)/'HUETE   ','YPRI SIG','NAL     ','1457P   ',
       $8*'        '/
        REAL LABEL/'    '/
        DATA PI/3.141592954/
        DOUBLE PRECISION DSEED
C
C       DEFINE AND COMPUTE ALL COEFFICIENTS
C       TEN SAMPLES PER SECOND
C
        T=1./10.
C
C       COEFFICIENTS FOR FIXED HIGH PASS FILTER
C
        AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
        AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
        BFHP0=(1./(1.+T/8.+T**2/320.))
        BFHP1=-(2./(1.+T/8.+T**2/320.))
        BFHP2=(1./(1.+T/8.+T**2/320.))
C
C       COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
C       IN THIS CASE,   F(LOWER)=0.04 HZ
```

```
C
      A=12.52096/40.82834
      B=1./40.82834
      C=11.00999/45.28317
      D=1./45.28317
      E=7.41498/57.57668
      F=1./57.57668
      A1=1.+A*T/2.+B*(T**2)/4.
      B1=-2.+B*(T**2/2.)
      C1=1.-A*T/2.+B*(T**2)/4.
      D1=1.+C*T/2.+D*(T**2)/4.
      E1=-2.+D*(T**2)/2.
      F1=1.-C*T/2.+D*(T**2)/4.
      G1=1.+E*T/2.+F*(T**2)/4.
      H1=-2.+F*(T**2)/2.
      I1=1.-E*T/2.+F*(T**2)/4.
C
C     CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
C     PASS FILTER WITH LOWER LIMIT 0.04 HZ"
C
      ASHP41=-(G1*(A1*E1+B1*D1)+(H1*A1*D1))/(G1*A1*D1)
      ASHP42=-(G1*(A1*F1+B1*E1+C1*D1)+H1*(A1*E1+B1*D1)+(I1*A1*D1))/
     $(G1*A1*D1)
      ASHP43=-(G1*(B1*F1+C1*E1)+H1*(A1*F1+B1*E1+C1*D1)+I1*(A1*E1+B1*D1))
     $/(G1*A1*D1)
      ASHP44=-(G1*C1*F1+H1*(B1*F1+C1*E1)+I1*(A1*F1+B1*E1+C1*D1))/(G1*
     $A1*D1)
      ASHP45=-(H1*C1*F1+I1*(B1*F1+C1*E1))/(G1*A1*D1)
      ASHP46=-(I1*C1*F1)/(G1*A1*D1)
      BSHP40=1./(G1*A1*D1)
      BSHP41=-6./(G1*A1*D1)
      BSHP42=15./(G1*A1*D1)
      BSHP43=-20./(G1*A1*D1)
      BSHP44=15./(G1*A1*D1)
      BSHP45=-6./(G1*A1*D1)
      BSHP46=1./(G1*A1*D1)
C
C     COEFFIECIENTS FOR SELECTABLE LOW PASS FILTER  WITH UPPER FREQ.
C     OF 0.6 HZ
C
      AA=1./0.03492
      BB=0.35804/0.03492
      CC=1./0.03492
      DD=1./0.02779
      EE=0.20696/0.02779
      FF=1./0.02779
      AA1=AA*(T**2)/4.
      BB1=2.*AA1
```

```
      CC1=AA1
      DD1=DD*(T**2)/4.
      EE1=2.*DD1
      FF1=DD1
      GG1=(1.+BB*T/2.+CC*(T**2)/4.)
      HH1=(-2.+CC*(T**2)/2.)
      II1=(1.-BB*T/2.+CC*(T**2)/4.)
      JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
      KK1=(-2.+FF*(T**2)/2.)
      LL1=(1.-EE*T/2.+FF*(T**2)/4.)
      ASLP61=-(GG1*KK1+HH1*JJ1)/(GG1*JJ1)
      ASLP62=-(GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)
      ASLP63=-(HH1*LL1+II1*KK1)/(GG1*JJ1)
      ASLP64=-(II1*LL1)/(GG1*JJ1)
      BSLP60=(AA1*DD1)/(GG1*JJ1)
      BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
      BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
      BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)
      BSLP64=(CC1*FF1)/(GG1*JJ1)
C
C     FINISHED COMPUTING COEFFIECIENTS, INITIALIZE STORAGE REGISTERS
C
      DO 200 J=1,3000
      TRU(J)=0.
      YO(J)=0.
      YPO(J)=0.
      ASQ(J)=0.
      SIG(J)=0.
  200 CONTINUE
C
C     STORAGE REGISTERS SET TO 0,
C     SET UP EQUATIONS FOR FILTER
C
      DO 100 I=1,3000
      I1=I-1
      I2=I-2
      I3=I-3
      I4=I-4
      I5=I-5
      I6=I-6
      IF(I1.LT.1) I1=I
      IF(I2.LT.1) I2=1
      IF(I3.LT.1) I3=1
      IF(I4.LT.1) I4=1
      IF(I5.LT.1) I5=1
      IF(I6.LT.1) I6=1
      ANOI=GGUBFS(DSEED)
C
```

```
C       THIS NOISE STATEMENT ADDS NOISE OF +/- 1 TO THE SIGNAL
C       IN ADDITION TO A SINUSOID OF F=10 HZ
C
        ANOISE=2.*(ANOI-0.5)
        SIG(I)=TRU(I)+SIN(2.*PI*FLOAT(I)+PHI2)+ANOISE
        TIME(I)=FLOAT(I)/600.
        YO(I)=BFHPO*SIG(I)+BFHP1*SIG(I1)+BFHP2*SIG(I2)+AFHP1*YO(I1)
       $+AFHP2*YO(I2)
C
C       THE NEXT COMMENTED STEP WAS USED IN TROUBLESHOOTING BY INPUTTING
C       THE SIGNAL AT VARIOUS STAGES OF THE FILTER
C       YO(I)=SIG(I)
        YP1=ASHP41*YPO(I1)+ASHP42*YPO(I2)+ASHP43*YPO(I3)+ASHP44*YPO(I4)
        YP2=ASHP45*YPO(I5)+ASHP46*YPO(I6)+BSHP40*YO(I)+BSHP41*YO(I1)
        YP3=BSHP44*YO(I4)+BSHP45*YO(I5)+BSHP46*YO(I6)+BSHP42*YO(I2)
       $+BSHP43*YO(I3)
        YPO(I)=YP1+YP2+YP3
C
C       THE NEXT COMMENTED STEP WAS USED FOR TROUBLESHOOTING BY INPUTTING
C       THE SIGNAL AT VARIOUS STAGES OF THE SILTER
C       YPO(I)=SIG(I)
        GP1=ASLP61*ASQ(I1)+ASLP62*ASQ(I2)+ASLP63*ASQ(I3)+ASLP64*ASQ(I4)
        GP2=BSLP60*YPO(I)+BSLP61*YPO(I1)+BSLP62*YPO(I2)
       $+BSLP63*YPO(I3)+BSLP64*YPO(I4)
        ASQ(I)=GP1+GP2
C
C
  100 CONTINUE
C
C       COMPUTATIONS FINISHED AND ANSWERS STORED IN ARRAYS, PLOT OUTPUT
C
C       CALL SUBROUTINE DRAW FOR FIRST GRAPH, A TIME SERIES
C       REPRESENTATIONOF THE INPUT SIGNAL TO THE PROGRAM
C
C       ONLY ONE PLOT ON THIS GRAPH, X AXIS WILL BE MAGNITUDE AND
C       LABELLED"MAGNITUDE" ON A LINEAR SCALE
C
C       Y AXIS WILL BE TIME AND LABELLED "MINUTES" ON A LINEAR SCALE
C
        CALL DRAW(3000,TIME,YO,0,0,LABEL,TITLA,0,0,0,0,0,0,5,4,1,LAST)
C
C       END OF FIRST PLOT, PLOT SECOND PLOT
C
C       SECOND PLOTWILL BE A TIME SERIES REPRESENTATION OF THE ASQ81
C       OUTPUT AS COMPUTED
C
C       PLOT WILL HAVE X AXIS LABELLED "MAGNITUDE",Y AXIS LABELLED
C       "MINUTES", BOTH ON A LINEAR SCALE
```

```
C
      CALL DRAW(3000,TIME,ASQ,0,0,LABEL,TITLB,0,0,0,0,0,0,5,4,1,LAST)
C
C     THIRD PLOT WILL BE A PLOT OF THE 'TRUE' SIGNAL VERSUS TIME
C     WITH SIGNAL ON THE X AXIS AND TIME ON THE Y AXIS
C
      CALL DRAW(3000,TIME,YPO,0,0,LABEL,TITLC,0,0,0,0,0,0,5,4,1,LAST)
C
C     FINISHED PLCTTING
C
      STOP
      END
/*
```

APPENDIX F

DIGITAL SOFTWARE FOR SIMULATION - CASCADE FORM (SINUSOIDS AS INPUT)

```
//HUETE JOB (1457,1106),' ',CLASS=B
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C                    MGN5 FORTRAN
C     THIS PROGRAM IS DESIGNED TO TEST THE ACTION OF THE PRELIMINARY
C     DIGITAL FILTER PROGRAM FOR THE ASQ-81 BY INTRODUCING SIMULATED
C     MAGNETIC SIGNALS INTO THE SYSTEM THROUGH THE USE OF SINUSOIDS
C     OF VARYING FREQUENCY, BOTH WITHIN AND OUTSIDE THE FREQUENCY
C     RANGE OF THE FILTER, WITH RANDOM NOISE ADDED
C
C     SET UP ARRAYS.  SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
C     OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
C     PROGRAM
C
      DIMENSION SIG(3000),ASQ(3000),TRU(3000),TIME(3000)
      DIMENSICN YO(3000),YPO(3000)
      DIMENSION XI(3000),XII(3000),XIII(3000),XIV(3000),XV(3000)
      REAL*8 DSEEC
      REAL*8 T,AFHP1,AFHP2,BFHP0,BFHP1,BFHP2,A,B,C,D,E,F
      REAL*8 A1,B1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1
      REAL*8 AA,BB,CC,DD,EE,FF,AA1,BB1,CC1,DD1,EE1,FF1,GG1,HH1,II1
      REAL*8 JJ1,KK1,LL1
      REAL*8 ASHP41,ASHP42,ASHP43,ASHP44,ASHP45,ASHP46,ASHP47
      REAL*8 ASLP61,ASLP62,ASLP63,ASLP64
      REAL*8 BSLP60,BSLP61,BSLP62,BSLP63,BSLP64
      REAL*8 TITLA(12)/'HUETE    ','INPUT SI','GNAL    ','1457P   ',
     $8*'        '/
      REAL*8 TITLB(12)/'HUETE    ','OUTPUT S','IGNAL   ','1457P   ',
     $8*'        '/
      REAL*8 TITLC(12)/'HUETE    ','TRUE SIG','NAL     ','1457P   ',
     $8*'        '/
      REAL*8 TITLD(12)/'HUETE    ','Y OUT SI','GNAL    ','1457P   ',
     $8*'        '/
      REAL*8 TITLE(12)/'HUETE    ','YPRI  S1','GNAL    ','1457P   ',
     $8*'        '/
      REAL LABEL/'    '/
      DATA PI/3.141592954/
      DOUBLE PRECISION DSEED
C
C     DEFINE AND COMPUTE ALL COEFFICIENTS
C     TEN SAMPLES PER SECOND
C
      T=1./10.
C
C     COEFFICIENTS FOR FIXED HIGH PASS FILTER
C
      AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
      AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
```

```
      BFHP0=(1./(1.+T/8.+T**2/320.))
      BFHP1=-(2./(1.+T/8.+T**2/320.))
      BFHP2=(1./(1.+T/3.+T**2/320.))
C
C     COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
C     IN THIS CASE,    F(LOWER)=0.04 HZ
C
      A=12.52096/40.82834
      B=1./40.82834
      C=11.00999/45.28317
      D=1./45.28317
      E=7.41498/57.57668
      F=1./57.57668
      A1=1.+A*T/2.+B*(T**2)/4.
      B1=-2.+B*(T**2/2.)
      C1=1.-A*T/2.+B*(T**2)/4.
      D1=1.+C*T/2.+D*(T**2)/4.
      E1=-2.+D*(T**2)/2.
      F1=1.-C*T/2.+D*(T**2)/4.
      G1=1.+E*T/2.+F*(T**2)/4.
      H1=-2.+F*(T**2)/2.
      I1=1.-E*T/2.+F*(T**2)/4.
C
C     CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
C     PASS FILTER WITH LOWER LIMIT 0.04 HZ"
C
      ASHP41=1./(A1*D1*G1)
      ASHP42=-(C1/A1)
      ASHP43=-(B1/A1)
      ASHP44=-(E1/D1)
      ASHP45=-(F1/D1)
      ASHP46=-(H1/G1)
      ASHP47=-(I1/G1)
C
C     COEFFIECIENTS FOR SELECTABLE LOW PASS FILTER  WITH UPPER FREQ.
C     OF 0.6 HZ
C
      AA=1./0.03492
      BB=0.35804/0.03492
      CC=1./0.03492
      DD=1./0.02779
      EE=0.20696/0.02779
      FF=1./0.02779
      AA1=AA*(T**2)/4.
      BB1=2.*AA1
      CC1=AA1
      DD1=DD*(T**2)/4.
      EE1=2.*DD1
```

```
      FF1=DD1
      GG1=(1.+BB*T/2.+CC*(T**2)/4.)
      HH1=(-2.+CC*(T**2)/2.)
      II1=(1.-BB*T/2.+CC*(T**2)/4.)
      JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
      KK1=(-2.+FF*(T**2)/2.)
      LL1=(1.-EE*T/2.+FF*(T**2)/4.)
      ASLP61=-(GG1*KK1+HH1*JJ1)/(GG1*JJ1)
      ASLP62=-(GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)
      ASLP63=-(HH1*LL1+II1*KK1)/(GG1*JJ1)
      ASLP64=-(II1*LL1)/(GG1*JJ1)
      BSLP60=(AA1*DD1)/(GG1*JJ1)
      BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)
      BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)
      BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)
      BSLP64=(CC1*FF1)/(GG1*JJ1)
C
C     FINISHED COMPUTING COEFFIECIENTS, INITIALIZE STORAGE REGISTERS
C
      DO 200 J=1,3000
      TRU(J)=0.
      YO(J)=0.
      YPO(J)=0.
      ASQ(J)=0.
      SIG(J)=0.
  200 CONTINUE
C
C     STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL
C
      DSEED = 1456.
      PHI1=GGUBFS(DSEED)*2.*PI
      PHI2=GGUBFS(DSEED)*2.*PI
      DO 100 I=1,3000
      I1=I-1
      I2=I-2
      I3=I-3
      I4=I-4
      I5=I-5
      I6=I-6
      IF(I1.LT.1) I1=I
      IF(I2.LT.1) I2=1
      IF(I3.LT.1) I3=1
      IF(I4.LT.1) I4=1
      IF(I5.LT.1) I5=1
      IF(I6.LT.1) I6=1
      TRU(I)=SIN(0.02*PI*FLOAT(I)+PHI1)
      ANOI=GGUBFS(DSEED)
C
```

```
C     THIS NOISE STATEMENT ADDS NOISE OF +/- 0.25 TO THE SIGNAL
C     IN ADDITION TO A SINUSOID OF F=10 HZ OF MAG. +/- 0.5
C
      ANOISE=0.5*(ANOI-0.5)
      SIG(I)=TRU(I)+0.5*SIN(2.*PI*FLOAT(I)+PHI2)+ANOISE
      TIME(I)=FLOAT(I)/600.
      YO(I)=BFHP0*SIG(I)+BFHP1*SIG(I1)+BFHP2*SIG(I2)+AFHP1*YO(I1)
     $+AFHP2*YO(I2)
      XI(I)=ASHP41*YO(I)+ASHP42*XI(I2)+ASHP43*XI(I1)
      XII(I)=XI(I)+XI(I2)-2.*XI(I1)
      XIII(I)=XII(I)+ASHP44*XIII(I1)+ASHP45*XIII(I2)
      XIV(I)=XIII(I)-2.*XIII(I1)+XIII(I2)
      XV(I)=XIV(I)+ASHP46*XV(I1)+ASHP47*XV(I2)
      YPO(I)=XV(I)+XV(I2)-2.*XV(I1)
      GP1=ASLP61*ASQ(I1)+ASLP62*ASQ(I2)+ASLP63*ASQ(I3)+ASLP64*ASQ(I4)
      GP2=BSLP60*YPO(I)+BSLP61*YPO(I1)+BSLP62*YPO(I2)
     $+BSLP63*YPO(I3)+BSLP64*YPO(I4)
      ASQ(I)=GP1+GP2
C
C
  100 CONTINUE
C
C     COMPUTATIONS FINISHED AND ANSWERS STORED IN ARRAYS, PLOT OUTPUT
C
C     CALL SUBROUTINE DRAW FOR FIRST GRAPH, A TIME SERIES
C     REPRESENTATIONOF THE INPUT SIGNAL TO THE PROGRAM
C
C     ONLY ONE PLOT ON THIS GRAPH, X AXIS WILL BE MAGNITUDE AND
C     LABELLED"MAGNITUDE" ON A LINEAR SCALE
C
C     Y AXIS WILL BE TIME AND LABELLED "MINUTES" ON A LINEAR SCALE
C
      CALL DRAW(3000,TIME,YO,0,0,LABEL,TITLD,0,0,0,0,0,0,5,4,1,LAST)
C
C     END OF FIRST PLOT, PLOT SECOND PLOT
C
C     SECOND PLOTWILL BE A TIME SERIES REPRESENTATION OF THE ASQ81
C     OUTPUT AS COMPUTED
C
C     PLOT WILL HAVE X AXIS LABELLED "MAGNITUDE",Y AXIS LABELLED
C     "MINUTES", BOTH ON A LINEAR SCALE
C
      CALL DRAW(3000,TIME,ASQ,0,0,LABEL,TITLB,0,0,0,0,0,0,5,4,1,LAST)
C
C     THIRD PLOT WILL BE A PLOT OF THE 'TRUE' SIGNAL VERSUS TIME
C     WITH SIGNAL ON THE X AXIS AND TIME ON THE Y AXIS
C
      CALL DRAW(3000,TIME,YPO,0,0,LABEL,TITLE,0,0,0,0,0,0,5,4,1,LAST)
```

```
      CALL DRAW(3000,TIME,SIG,0,0,LABEL,TITLA,0,0,0,0,0,0,5,4,1,LAST)      APP04000
      CALL DRAW(3000,TIME,TRU,0,0,LABEL,TITLC,0,0,0,0,0,0,5,4,1,LAST)      APP04010
C                                                                          APP04020
C     FINISHED PLOTTING                                                    APP04030
C                                                                          APP04040
      STOP                                                                 APP04050
      END                                                                  APP04060
/*                                                                         APP04070
```

APPENDIX G

DIGITAL SOFTWARE FOR COMPUTATION OF SYSTEM AMPLITUDE VERSUS FREQUENCY

```
//HUETE JOB (1457,1106),' ',CLASS=B
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C                   MGN4 FORTRAN
C     THIS PROGRAM IS DESIGNED TO INPUT VARIOUS FREQUENCIES INTO THE
C     DIGITAL FILTER PROGRAM FOR THE ASQ-81 AND OBTAIN THE DB LOSS
C     CHARACTERISTIC FOR COMPARISON WITH MEASURED DB LOSSES FOR THE
C     ASQ 81 MAGNETOMETER.  A SINGLE FREQUENCY SIGNAL WILL BE INPUTTED
C     AND THE RMS OUTPUT DIVIDED BY THE RMS INPUT TO DETERMINE
C     ATTENUATION
C     SET UP ARRAYS.  SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
C     OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
C     PROGRAM
C
      DIMENSION SIG(3000),ASQ(3000),TRU(3000),TIME(3000)
      DIMENSION YO(3000),YPO(3000),FREQ(20)
      DIMENSION XI(3000),XII(3000),XIII(3000),XIV(3000),XV(3000)
      REAL*8 DSEED
      REAL*8 T,AFHP1,AFHP2,BFHP0,BFHP1,BFHP2,A,B,C,D,E,F
      REAL*8 A1,B1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1
      REAL*8 AA,BB,CC,DD,EE,FF,AA1,BB1,CC1,DD1,EE1,FF1,GG1,HH1,II1
      REAL*8 JJ1,KK1,LL1
      REAL*8 ASHP41,ASHP42,ASHP43,ASHP44,ASHP45,ASHP46,ASHP47
      REAL*8 ASLP61,ASLP62,ASLP63,ASLP64
      REAL*8 BSLP60,BSLP61,BSLP62,BSLP63,BSLP64
      DATA PI/3.141592954/
      DOUBLE PRECISION DSEED,SUMSQ,SMSQT,RATIO
C
C     DEFINE AND COMPUTE ALL COEFFICIENTS
C     TEN SAMPLES PER SECOND
C
      T=1./10.
C
C     COEFFICIENTS FOR FIXED HIGH PASS FILTER
C
      AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
      AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
      BFHP0=(1./(1.+T/8.+T**2/320.))
      BFHP1=-(2./(1.+T/8.+T**2/320.))
      BFHP2=(1./(1.+T/8.+T**2/320.))
C
C     COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
C     IN THIS CASE,   F(LOWER)=0.04 HZ
C
      A=12.52096/40.82834
      B=1./40.82834
      C=11.00999/45.28317
      D=1./45.28317
```

```
      E=7.41498/57.57668
      F=1./57.57668
      A1=1.+A*T/2.+B*(T**2)/4.
      B1=-2.+B*(T**2/2.)
      C1=1.-A*T/2.+B*(T**2)/4.
      D1=1.+C*T/2.+D*(T**2)/4.
      E1=-2.+D*(T**2)/2.
      F1=1.-C*T/2.+D*(T**2)/4.
      G1=1.+E*T/2.+F*(T**2)/4.
      H1=-2.+F*(T**2)/2.
      I1=1.-E*T/2.+F*(T**2)/4.
C
C     CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
C     PASS FILTER WITH LOWER LIMIT 0.04 HZ"
C
      ASHP41=1./(A1*D1*G1)
      ASHP42=-(C1/A1)
      ASHP43=-(B1/A1)
      ASHP44=-(E1/D1)
      ASHP45=-(F1/D1)
      ASHP46=-(H1/G1)
      ASHP47=-(I1/G1)
C
C     COEFFIECIENTS FOR SELECTABLE LOW PASS FILTER  WITH UPPER FREQ.
C     OF 0.6 HZ
C
      AA=1./0.03492
      BB=0.35804/0.03492
      CC=1./0.03492
      DD=1./0.02779
      EE=0.20696/0.02779
      FF=1./0.02779
      AA1=AA*(T**2)/4.
      BB1=2.*AA1
      CC1=AA1
      DD1=DD*(T**2)/4.
      EE1=2.*DD1
      FF1=DD1
      GG1=(1.+BB*T/2.+CC*(T**2)/4.)
      HH1=(-2.+CC*(T**2)/2.)
      II1=(1.-BB*T/2.+CC*(T**2)/4.)
      JJ1=(1.+EE*T/2.+FF*(T**2)/4.)
      KK1=(-2.+FF*(T**2)/2.)
      LL1=(1.-EE*T/2.+FF*(T**2)/4.)
      ASLP61=-(GG1*KK1+HH1*JJ1)/(GG1*JJ1)
      ASLP62=-(GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)
      ASLP63=-(HH1*LL1+II1*KK1)/(GG1*JJ1)
      ASLP64=-(II1*LL1)/(GG1*JJ1)
```

```
      BSLP60=(AA1*DD1)/(GG1*JJ1)                                          APP05060
      BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)                                  APP05070
      BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)                          APP05080
      BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)                                  APP05090
      BSLP64=(CC1*FF1)/(GG1*JJ1)                                          APP05100
C                                                                         APP05110
C     FINISHED COMPUTING COEFFIECIENTS, INITIALIZE STORAGE REGISTERS      APP05120
C                                                                         APP05130
      DO 200 J=1,3000                                                     APP05140
      TRU(J)=0.                                                           APP05150
      YO(J)=0.                                                            APP05160
      YPO(J)=0.                                                           APP05170
      ASQ(J)=0.                                                           APP05180
      SIG(J)=0.                                                           APP05190
  200 CONTINUE                                                            APP05200
C                                                                         APP05210
C     STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL          APP05220
C                                                                         APP05230
      DSEED = 1456.                                                       APP05240
      PHI1=GGUBFS(DSEED)*2.*PI                                            APP05250
      PHI2=GGUBFS(DSEED)*2.*PI                                            APP05260
      FREQ(1)=0.01                                                        APP05270
      FREQ(2)=0.02                                                        APP05280
      FREQ(3)=0.03                                                        APP05290
      FREQ(4)=0.04                                                        APP05300
      FREQ(5)=0.05                                                        APP05310
      FREQ(6)=0.06                                                        APP05320
      FREQ(7)=0.07                                                        APP05330
      FREQ(8)=0.08                                                        APP05340
      FREQ(9)=0.09                                                        APP05350
      FREQ(10)=0.10                                                       APP05360
      FREQ(11)=0.20                                                       APP05370
      FREQ(12)=0.30                                                       APP05380
      FREQ(13)=0.40                                                       APP05390
      FREQ(14)=0.50                                                       APP05400
      FREQ(15)=0.60                                                       APP05410
      FREQ(16)=0.70                                                       APP05420
      FREQ(17)=0.80                                                       APP05430
      FREQ(18)=0.90                                                       APP05440
      FREQ(19)=1.0                                                        APP05450
      FREQ(20)=0.15                                                       APP05460
      DO 300 II=1,20                                                      APP05470
      DO 100 I=1,3000                                                     APP05480
      I1=I-1                                                              APP05490
      I2=I-2                                                              APP05500
      I3=I-3                                                              APP05510
      I4=I-4                                                              APP05520
      I5=I-5                                                              APP05530
```

```
      I6=I-6
      IF(I1.LT.1) I1=I
      IF(I2.LT.1) I2=1
      IF(I3.LT.1) I3=1
      IF(I4.LT.1) I4=1
      IF(I5.LT.1) I5=1
      IF(I6.LT.1) I6=1
      TRU(I)=SIN(0.2*PI*FREQ(II)*FLOAT(I))
      SIG(I)=TRU(I)
      TIME(I)=FLOAT(I)/600.
C
C     THE FIRST STAGE OF THE FILTER PROGRAM IS COMMENTED OUT BECAUSE IN
C     THIS VERSION OF THE PROGRAM THE FIXED HIGH PASS FILTER IS NOT
C     INCLUDED SO AS TO ENABLE COMPARISON WITH DATA FURNISHED BY
C     TEXAS INSTRUMENTS, INC.  BY REMOVING THE COMMENT CHARACTERS, AND
C     COMMENTING OUT THE STEP FOLLOWING THEM (  YO(I)=SIG(I)      ), THE
C     PROGRAM CAN BE MADE TO INCLUDE THE FIXED HIGH PASS FILTER
C
C     YO(I)=BFHPO*SIG(I)+BFHP1*SIG(I1)+BFHP2*SIG(I2)+AFHP1*YO(I1)
C    $+AFHP2*YO(I2)
      YO(I)=SIG(I)
      XI(I)=ASHP41*YO(I)+ASHP42*XI(I2)+ASHP43*XI(I1)
      XII(I)=XI(I)+XI(I2)-2.*XI(I1)
      XIII(I)=XII(I)+ASHP44*XIII(I1)+ASHP45*XIII(I2)
      XIV(I)=XIII(I)-2.*XIII(I1)+XIII(I2)
      XV(I)=XIV(I)+ASHP46*XV(I1)+ASHP47*XV(I2)
      YPO(I)=XV(I)+XV(I2)-2.*XV(I1)
      GP1=ASLP61*ASQ(I1)+ASLP62*ASQ(I2)+ASLP63*ASQ(I3)+ASLP64*ASQ(I4)
      GP2=BSLP60*YPO(I)+BSLP61*YPO(I1)+BSLP62*YPO(I2)
     $+BSLP63*YPO(I3)+BSLP64*YPO(I4)
      ASQ(I)=GP1+GP2
C
C
  100 CONTINUE
C
C     THE FOLLOWING SECTION COMPUTES THE AVERAGE VALUES OF THE OUTPUT
C     AND CONVERTS TO DB ATTENUATION
C
      SUMSQ=0.0
      SMSQT=0.0
      DO 301 JJ=2000,3000
      SUMSQ=SUMSQ+(ASQ(JJ)**2)
      SMSQT=SMSQT+(TRU(JJ)**2)
  301 CONTINUE
      RATIO=SUMSQ/SMSQT
      DBLOSS=10.*DLOG10(RATIO)
      WRITE(6,4001)FREQ(II),DBLOSS
 4001 FORMAT(1X,'FREQUENCY = ',F10.2,'  DB LOSS =', F10.5)
```

```
C                                                                    APP06020
C     FINISHED PLOTTING                                              APP06030
C                                                                    APP06040
  300 CONTINUE                                                       APP06050
      STOP                                                           APP06060
      END                                                            APP06070
/*                                                                   APP06080
```

APPENDIX H

DIGITAL SOFTWARE FOR SIMULATION (ANDERSON FUNCTIONS AS INPUT)

```
//HUETE JOB (1457,1106),' ',CLASS=B
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C                    MGN6 FORTRAN
C     THIS PROGRAM IS DESIGNED TO TEST THE ACTION OF THE PRELIMINARY
C     DIGITAL FILTER PROGRAM FOR THE ASQ-81 BY INTRODUCING SIMULATED
C     SUBMARINE SIGNALS INTO THE SYSTEM THROUGH THE USE OF ANDERSON
C     FUNCTIONS OF VARYING RANGES AND SPEEDS.
C
C
C     SET UP ARRAYS.  SIG() IS THE SIGNAL, ASQ() IS THE PROGRAM
C     OUTPUT, TRU() IS THE SIGNAL WITHIN THE FREQUENCY RANGE OF THE
C     PROGRAM
C
      DIMENSION SIG(3000),ASQ(3000),TRU(3000),TIME(3000)
      DIMENSION YO(3000),YPO(3000)
      DIMENSION XI(3000),XII(3000),XIII(3000),XIV(3000),XV(3000)
      REAL*8 DSEED
      REAL*8 T,AFHP1,AFHP2,BFHP0,BFHP1,BFHP2,A,B,C,D,E,F
      REAL*8 A1,B1,C1,D1,E1,F1,G1,H1,I1,J1,K1,L1
      REAL*8 AA,BB,CC,DD,EE,FF,AA1,BB1,CC1,DD1,EE1,FF1,GG1,HH1,II1
      REAL*8 JJ1,KK1,LL1
      REAL*8 ASHP41,ASHP42,ASHP43,ASHP44,ASHP45,ASHP46,ASHP47
      REAL*8 ASLP61,ASLP62,ASLP63,ASLP64
      REAL*8 BSLP60,BSLP61,BSLP62,BSLP63,BSLP64
      REAL*8 TITLA(12)/'HUETE   ','INPUT SI','GNAL    ','1457P   ',
     $8*'        '/
      REAL*8 TITLB(12)/'HUETE   ','OUTPUT S','IGNAL   ','1457P   ',
     $8*'        '/
      REAL*8 TITLD(12)/'HUETE   ','Y OUT SI','GNAL    ','1457P   ',
     $8*'        '/
      REAL*8 TITLE(12)/'HUETE   ','YPRI  SI','GNAL    ','1457P   ',
     $8*'        '/
      REAL LABEL/'    '/
      DATA PI/3.141592954/
      DOUBLE PRECISION DSEED,BETA,NORM
C
C     DEFINE AND COMPUTE ALL COEFFICIENTS
C     TEN SAMPLES PER SECOND
C
      T=0.125
C
C     COEFFICIENTS FOR FIXED HIGH PASS FILTER
C
      AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))
      AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))
      BFHP0=(1./(1.+T/8.+T**2/320.))
      BFHP1=-(2./(1.+T/8.+T**2/320.))
```

```
      BFHP2=(1./(1.+T/8.+T**2/320.))
C
C     COEFFICIENTS FOR SELECTABLE HIGH PASS FILTER
C     IN THIS CASE,    F(LOWER)=0.04 HZ
C
      A=12.52096/40.82834
      B=1./40.82834
      C=11.00999/45.28317
      D=1./45.28317
      E=7.41498/57.57668
      F=1./57.57668
      A1=1.+A*T/2.+B*(T**2)/4.
      B1=-2.+B*(T**2/2.)
      C1=1.-A*T/2.+B*(T**2)/4.
      D1=1.+C*T/2.+D*(T**2)/4.
      E1=-2.+D*(T**2)/2.
      F1=1.-C*T/2.+D*(T**2)/4.
      G1=1.+E*T/2.+F*(T**2)/4.
      H1=-2.+F*(T**2)/2.
      I1=1.-E*T/2.+F*(T**2)/4.
C
C     CODE IS "ASHP41" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
C     PASS FILTER WITH LOWER LIMIT 0.04 HZ"
C
      ASHP41=1./(A1*D1*G1)
      ASHP42=-(C1/A1)
      ASHP43=-(B1/A1)
      ASHP44=-(E1/D1)
      ASHP45=-(F1/D1)
      ASHP46=-(H1/G1)
      ASHP47=-(I1/G1)
C
C     COEFFIECIENTS FOR SELECTABLE LOW PASS FILTER  WITH UPPER FREQ.
C     OF  0.6 HZ
C
      AA=1./0.03492
      BB=0.35804/0.03492
      CC=1./0.03492
      DD=1./0.02779
      EE=0.20696/0.02779
      FF=1./0.02779
      AA1=AA*(T**2)/4.
      BB1=2.*AA1
      CC1=AA1
      DD1=DD*(T**2)/4.
      EE1=2.*DD1
      FF1=DD1
      GG1=(1.+BB*T/2.+CC*(T**2)/4.)
```

```
      HH1=(-2.+CC*(T**2)/2.)                                            APP07070
      II1=(1.-BB*T/2.+CC*(T**2)/4.)                                     APP07080
      JJ1=(1.+EE*T/2.+FF*(T**2)/4.)                                     APP07090
      KK1=(-2.+FF*(T**2)/2.)                                            APP07100
      LL1=(1.-EE*T/2.+FF*(T**2)/4.)                                     APP07110
      ASLP61=-(GG1*KK1+HH1*JJ1)/(GG1*JJ1)                               APP07120
      ASLP62=-(GG1*LL1+HH1*KK1+II1*JJ1)/(GG1*JJ1)                       APP07130
      ASLP63=-(HH1*LL1+II1*KK1)/(GG1*JJ1)                               APP07140
      ASLP64=-(II1*LL1)/(GG1*JJ1)                                       APP07150
      BSLP60=(AA1*DD1)/(GG1*JJ1)                                        APP07160
      BSLP61=(AA1*EE1+BB1*DD1)/(GG1*JJ1)                                APP07170
      BSLP62=(AA1*FF1+BB1*EE1+CC1*DD1)/(GG1*JJ1)                        APP07180
      BSLP63=(BB1*FF1+CC1*EE1)/(GG1*JJ1)                                APP07190
      BSLP64=(CC1*FF1)/(GG1*JJ1)                                        APP07200
C                                                                       APP07210
C     FINISHED COMPUTING COEFFIECIENTS, INITIALIZE STORAGE REGISTERS    APP07220
C                                                                       APP07230
      DO 200 J=1,3000                                                   APP07240
      TRU(J)=0.                                                         APP07250
      YO(J)=0.                                                          APP07260
      YPO(J)=0.                                                         APP07270
      ASQ(J)=0.                                                         APP07280
      SIG(J)=0.                                                         APP07290
  200 CONTINUE                                                          APP07300
C                                                                       APP07310
C     STORAGE REGISTERS SET TO 0, EQUATIONS FOR SIMULATED SIGNAL        APP07320
C                                                                       APP07330
      DO 100 I=1,2400                                                   APP07340
      I1=I-1                                                            APP07350
      I2=I-2                                                            APP07360
      I3=I-3                                                            APP07370
      I4=I-4                                                            APP07380
      I5=I-5                                                            APP07390
      I6=I-6                                                            APP07400
      IF(I1.LT.1) I1=I                                                  APP07410
      IF(I2.LT.1) I2=1                                                  APP07420
      IF(I3.LT.1) I3=1                                                  APP07430
      IF(I4.LT.1) I4=1                                                  APP07440
      IF(I5.LT.1) I5=1                                                  APP07450
      IF(I6.LT.1) I6=1                                                  APP07460
C                                                                       APP07470
C     THIS SIGNAL STATEMENT INPUTS A 180 KNOTS AIRCRAFT AT A CPA        APP07480
C     RANGE OF  400 FEET.  THIS IS TO TEST THE OPERATION OF THE         APP07490
C     SELCETABLE LOW PASS FILTER      SECOND ANDERSON FUNCTION          APP07500
C     8 HZ FILTER                                                       APP07510
C                                                                       APP07520
      TIME(I)=FLOAT(I)/480.                                             APP07530
      BETA=18000.*(TIME(I)-2.5)/400.                                    APP07540
```

```
      BETP=DSQRT((BETA**2)+1.)
C
C     THE FACTOR "NORM" IS A NORMALIZATION FACTOR TO MAKE THE INPUT
C     SIGNAL +/- AMPLITUDE 1
C
      NORM=1.7469281
C
C     THIS SIGNAL IS THE FIRST ANDERSON FUNCTION.  TO OBTAIN THE SECOND
C     ANDERSON FUNCTION, MULTIPLY BY BETA, AND AGAIN TO OBTAIN THE
C     THIRD ANDERSON FUNCTION
C
      SIG(I)=NORM*((1./BETP)**5)
      YO(I)=BFHP0*SIG(I)+BFHP1*SIG(I1)+BFHP2*SIG(I2)+AFHP1*YO(I1)
     $+AFHP2*YO(I2)
      XI(I)=ASHP41*YO(I)+ASHP42*XI(I2)+ASHP43*XI(I1)
      XII(I)=XI(I)+XI(I2)-2.*XI(I1)
      XIII(I)=XII(I)+ASHP44*XIII(I1)+ASHP45*XIII(I2)
      XIV(I)=XIII(I)-2.*XIII(I1)+XIII(I2)
      XV(I)=XIV(I)+ASHP46*XV(I1)+ASHP47*XV(I2)
      YPO(I)=XV(I)+XV(I2)-2.*XV(I1)
      GP1=ASLP61*ASQ(I1)+ASLP62*ASQ(I2)+ASLP63*ASQ(I3)+ASLP64*ASQ(I4)
      GP2=BSLP60*YPO(I)+BSLP61*YPO(I1)+BSLP62*YPO(I2)
     $+BSLP63*YPO(I3)+BSLP64*YPO(I4)
      ASQ(I)=GP1+GP2
C
C
  100 CONTINUE
C
C     COMPUTATIONS FINISHED AND ANSWERS STORED IN ARRAYS, PLOT OUTPUT
C
C     CALL SUBROUTINE DRAW FOR FIRST GRAPH, A TIME SERIES
C     REPRESENTATIONOF THE INPUT SIGNAL TO THE PROGRAM
C
C     ONLY ONE PLOT ON THIS GRAPH, X AXIS WILL BE MAGNITUDE AND
C     LABELLED"MAGNITUDE" ON A LINEAR SCALE
C
C     Y AXIS WILL BE TIME AND LABELLED "MINUTES" ON A LINEAR SCALE
C
      CALL DRAW(2400,TIME,YO,0,0,LABEL,TITLD,0,0,0,0,0,0,10,4,1,LAST)
C
C     END OF FIRST PLOT, PLOT SECOND PLOT
C
C     SECOND PLOTWILL BE A TIME SERIES REPRESENTATION OF THE ASQ81
C     OUTPUT AS COMPUTED
C
C     PLOT WILL HAVE X AXIS LABELLED "MAGNITUDE",Y AXIS LABELLED
C     "MINUTES", BOTH ON A LINEAR SCALE
C
```

```
      CALL DRAW(2400,TIME,ASQ,0,0,LABEL,TITLB,0,0,0,0,0,0,10,4,1,LAST)   APP08030
C                                                                        APP08040
C     THIRD PLOT WILL BE A PLOT OF THE 'TRUE' SIGNAL VERSUS TIME         APP08050
C     WITH SIGNAL ON THE X AXIS AND TIME ON THE Y AXIS                   APP08060
C                                                                        APP08070
      CALL DRAW(2400,TIME,YPO,0,0,LABEL,TITLE,0,0,0,0,0,0,10,4,1,LAST)   APP08080
      CALL DRAW(2400,TIME,SIG,0,0,LABEL,TITLA,0,0,0,0,0,0,10,4,1,LAST)   APP08090
C                                                                        APP08100
C     FINISHED PLOTTING                                                  APP08110
C                                                                        APP08120
      STOP                                                               APP08130
      END                                                                APP08140
/*                                                                       APP08150
```

APPENDIX I

DIGITAL FILTERING SOFTWARE

```
//HUETE JOB (1457,0165),'HUETE      SMC 2740',CLASS=G                  APP08180
//*MAIN LINES=(65)                                                      APP08190
// EXEC FRTXCLGP,PARM.LKED='LIST,MAP,XREF',REGION.GO=2048K             APP08200
//FORT.SYSIN DD *                                                       APP08210
      INTEGER*2 IN(16)                                                  APP08220
C      ARRAY 'IN' IS USED IN READING DATA FROM TAPE                     APP08230
      COMPLEX*8 XX(8192)                                                APP08240
      REAL*4 ZZ(8192),YY(8192),XXP(8192)                                APP08250
C      THE ABOVE COMPLEX*8 ARRAYS ARE USED TO ORDER INPUT DATA AND      APP08260
C      INITIALLY REPRESENT VOLTAGE - TIME SERIES INFORMATION.           APP08270
C                                                                       APP08280
C     THE NEXT THREE LINES ARE ARRAYS NEEDED FOR DATA TAPE READING AND  APP08290
C     CONVERSION TO TOTAL FIELD FLUCTUATION TIME SERIES                 APP08300
C                                                                       APP08310
      DIMENSION TIME(8192),FREQ(8192),WORK(24576),FRQ2(8192)            APP08320
      DIMENSION ZX1(8192),ZY1(8192)                                     APP08330
      DIMENSION ZZX1(24576),ZZY1(24576),ZZV1(24576)                     APP08340
C                                                                       APP08350
C     THE FOLLOWING LINES CONTAIN ARRAYS NEEDED FOR SIGNAL INPUT TO THE APP08360
C     FILTER, SIGNAL PROCESSING WITHIN THE FILTER, AND COEFFICIENTS     APP08370
C     USED BY THE FILTER PROGRAM                                        APP08380
C                                                                       APP08390
      DIMENSION TIME2(24576),OUTFLD(24576),CLFLD(24576)                 APP08400
      REAL*8 AH(4),BH(4),CH(4),DH(4),EH(4),FH(4)                        APP08410
      REAL*8 AL(3),BL(3),CL(3),DL(3),EL(3),FL(3),FRQH(4),FRQL(3)        APP08420
      REAL*8 T,AFHP1,AFHP2,BFHP0,BFHP1,BFHP2,A1,B1,C1,D1,E1,F1,G1,H1    APP08430
      REAL*8 I1,J1,K1,L1,ASHP1,ASHP2,ASHP3,ASHP4,ASHP5,ASHP6,ASHP7      APP08440
      REAL*8 ASLP1,ASLP2,ASLP3,ASLP4,BSLP0,BSLP1,BSLP2,BSLP3,BSLP4      APP08450
      REAL*8 SIG2,SIG1,SIG,YO2,YO1,YO,XI2,XI1,XI,XIII2,XIII1,XIII,XV2   APP08460
      REAL*8 XV1,XV,YP04,YP03,YP02,YP01,YP0,OUTFD4,OUTFD3,OUTFD2        APP08470
      REAL*8 OUTFD1,GP1,GP2,XII,XIV                                     APP08480
C      THE 'Z' ARRAYS REPRESENT FREQUENCY DOMAIN (FF TRANSFORMED)       APP08490
C      MAGNITUDE DATA AND ARE EVENTUALLY CONVERTED TO POWER SPECTRAL    APP08500
C      DENSITY INFORMATION. ZX1,ZY1,ZV1,AND ZT1 REPRESENT MAGNITUDE     APP08510
C      VALUES.                                                          APP08520
C                                                                       APP08530
C     THE NEXT LINES CONTAIN CONSTANTS AND ARRAYS USED IN PLOTTING      APP08540
C     THE OUTPUT                                                        APP08550
C                                                                       APP08560
      INTEGER K,I4,I5,Q                                                 APP08570
      REAL SUMX,SUMY,SUMZ,AVE1                                          APP08580
      REAL CONSTX                                                       APP08590
      INTEGER*4 ITB(12)/12*0/                                           APP08600
      REAL*4    RTB(28)/28*0.0/                                         APP08610
      REAL ALAB(4)/'    ','    ','    ','    '/                         APP08620
      REAL*8 TITLE(12)                                                  APP08630
      EQUIVALENCE(TITLE(1),RTB(5))                                      APP08640
C                                                                       APP08650
```

```
C     SET VARIABLES EQUAL TO ZERO
C
      DATA XX/8192*(0.,0.)/
      DATA ZZ,YY/16384*0./
      DATA ZX1/8192*0./
      DATA TIME,FREQ/16384*0./
      K=0
      I4=1
      I5=1
      CONSTX=0.0
      SUMX=0.0
      SUMY=0.0
      SUMZ=0.0
      AVE1=0.0
      XORIGP=0.0
      XMAXP=0.0
C
C     SET STORAGE REGISTERS TO ZERO.  STORAGE REGISTERS ARE USED VICE
C     INTERMEDIATE OUTPUT ARRAYS IN ORDER TO CUT DOWN THE AMOUNT OF
C     ARRAY STORAGE REQUIRED BY THE PROGRAM AND TO RETAIN "MEMORY"
C     OF PREVIOUS VALUES FOR COMPUTATIONAL USE IN ORDER TO ELIMINATE
C     THE "START UP" LAG OF THE OUTPUT VALUES
C
      SIG2=0.0
      SIG1=0.0
      YO2=0.0
      YO1=0.0
      XI2=0.0
      XI1=0.0
      XIII2=0.0
      XIII1=0.0
      XV2=0.0
      XV1=0.0
      YPO4=0.0
      YPO3=0.0
      YPO2=0.0
      YPO1=0.0
      OUTFD4=0.0
      OUTFD3=0.0
      OUTFD2=0.0
      OUTFD1=0.0
C
C     THE FOLLOWING SEVERAL STEPS WOULD BE USED IF THE INPUT TO THE
C     FILTER PROGRAM WERE THREE MUTUALLY PERPENDICULAR COIL SENSORS.
C     SINCE THE INPUT IS A SINGLE COIL SENSOR ORIENTED ALONG THE
C     EARTH'S FIELD, THESE STEPS ARE NOT NECESSARY, BUT ARE RETAINED
C     AS REFERENCE
C
```

```
      TWOPI=6.2831853
      COS60=COS(TWOPI/6.)
      COS30=COS(TWOPI/12.)
      D=16.75*TWOPI/360.
      COSD=COS(D)
      COSD1=COS((90. - D)*TWOPI/360.)
C        D IS THE DECLINATION OR MAGNETIC VARIATION AT THE MAGNETOMETER
C        SITE.
C
C     SET ARRAYS TO ZERO
C
      DO 31 IN1=1,24576
      ZZX1(IN1)=0.0
      ZZY1(IN1)=0.0
      ZZV1(IN1)=0.0
      TIME2(IN1)=0.0
   31 CONTINUE
C        THE NEXT FIVE LINES SERVE AS A TIME DELAY IN STARTING THE
C        DATA ANALYSIS
      ISEC=10
      ITL=ISEC*64
      DO 55 JJ=1,ITL
      CALL RD(20,IN,200,IREC,IRR)
   55 CONTINUE
      IFRAME=8192
      NR=1
      FNR=FLOAT(NR)
      DO 200 IM=1,NR
      XORIGP=XMAXP
      DO 70 IL=1,3
C      THE DO LOOP ENDING WITH STATEMENT 70 ENABLES THE PROGRAM TO
C      PROCESS A LARGE AMOUNT OF DATA BY REPEATING THE PROCESS IN
C      BLOCKS. THE DATA POINTS FROM EACH RUN THROUGH THE DO LOOP ARE
C      ADDED TOGETHER AND EVENTUALLY AVERAGED BY THE NUMBER OF RUNS
C      THROUGH THE DO LOOP.
C      'NR' REPRESENTS THE NUMBER OF DATA SEQUENCES TO BE AVERAGED.
C      1 SEQUENCE CURRENTLY EQUALS 24576 DATA POINTS OR THREE SETS
C      OF 128 SECONDS OF DATA.
C
C        THE DO LOOP ENDING WITH 60 READS THE DATA FROM THE PCM FRAME
C        STRIPS OUT THE SYNC CODE, AND SORTS OUT THE DATA BY COIL
C        CHANNEL
      DO 60 JJ=1,IFRAME
      CALL RD(20,IN,1000,IREC,IRR)
      XX(JJ)=IN(2)
      YY(JJ)=IN(3)
      ZZ(JJ)=IN(4)
   60 CONTINUE
```

```
C     WRITE(6,200)IRR,IREC
C 200 FORMAT(10X,'IRR=',I6,5X,'IREC=',I6,/)
C       THE FOLLOWING SECTION GENERATES THE TIME AND FREQUENCY
C       ARRAYS AND NORMALIZES THE INPUT PCM DATA TO VOLTAGE FORM
C       IN PREPARATION FOR FAST FOURIER TRANSFORM TO THE FREQUENCY
C       DOMAIN.
      N=8192
      FN=FLOAT(N)
      DELTAT=1./64.
      DELTAF=1./(FN*DELTAT)
      DO 20 J=1,N
      TIME(J)=DELTAT*FLOAT(J)
      FREQ(J)=DELTAF*FLOAT(J)
      XX(J)=(XX(J)-2048.)*5./2048.
      XX(J)=REAL(XX(J))
      XXP(J)=XX(J)
      YY(J)=(YY(J)-2048.)*5./2048.
      ZZ(J)=(ZZ(J)-2048.)*5./2048.
C
C     IN THIS USE OF THE PROGRAM,
C       'XX' IS THE COIL ANTENNA DATA, 'YY' IS THE ASQ-81 DATA
C       'ZZ' IS THE SCHONSTEDT COIL DATA, AND 'TF' IS THE
C       TOTAL GEOMAGNETIC FIELD VECTOR.
C     IF THREE MUTUALLY PERPENDICULAR COIL SENSORS ARE USED, THIS
C     WILL NOT BE TRUE.  SEE REFERENCE 9 FOR HOW TO HANDLE THIS
C
   20 CONTINUE
      DO 21 J=1,N
      FRQ2(J)=ALOG10(FREQ(J))
   21 CONTINUE
C       THE NEXT FOUR STATEMENTS PERFORM AN FFT ON THE INPUT
C       TIME SERIES DATA. SEE THE WRITEUP ON 'FOURT' FOR
C       FURTHER INFORMATION.
      CALL FOURT(XX,N,1,-1,0,WORK)
C      THE NEXT BLOCK OF STATEMENTS APPLY THE SYSTEM (VOLTAGE TO
C      B-FIELD) TRANSFER FUNCTION TO THE TRANSFORMED FREQUENCY
C      DOMAIN DATA.  THIS BLOCK ENDS AT STATEMENT 9.
C      THE TRANSFER FUNCTION CONVERTS VOLTS TO NANOTESLAS (GAMMAS).
C      ***WARNING*** THIS TRANSFER FUNCTIONS YIELDS AN INACCURATE
C      PHASE.  USE A DIFFERENT TRANSFER FUNCTION IF PHASE INFORMATION
C      NEEDED.
      DO 9 L=1,N
      FRQ=FREQ(L)
      IF(FRQ.LE.25.)GO TO 1
      XX(L)=XX(L)/28.
      GO TO 8
    1 IF(FRQ.LE.15.)GO TO 2
      XX(L)=XX(L)/(105.5-3.14*FRQ)
```

```
      GO TO 8
    2 IF(FRQ.LE.10.)GO TO 3
      XX(L)=XX(L)/(5.958*FRQ-30.97)
      GO TO 8
    3 IF(FRQ.LE.7.5)GO TO 4
      XX(L)=XX(L)/(3.492*FRQ-6.31)
      GO TO 8
    4 IF(FRQ.LE.5.)GO TO 5
      XX(L)=XX(L)/(2.6311*FRQ+0.14667)
      GO TO 8
    5 IF(FRQ.LE.3.)GO TO 6
      XX(L)=XX(L)/(2.6311*FRQ+0.14667)
      GO TO 8
    6 XX(L)=XX(L)/(2.72*FRQ)
      GO TO 8
    8 CONTINUE
    9 CONTINUE
      CALL FOURT(XX,N,1,1,1,WORK)
      DO 57 J=1,N
      XX(J)=XX(J)/FN
   57 CONTINUE
C     WRITE(6,600)(XX(I),I=1,100)
C600  FORMAT(1X,F20.4,4X,F20.4)
C      THE FOLLOWING BLOCK TAKES THE MAGNITUDE OF THE COMPLEX VALUES
      DO 56 I3=1,N
      ZX1(I3)=CABS(XX(I3))
   56 CONTINUE
      IF(K.NE.0) GO TO 36
      DO 66 IS=8048,8192
      SUMX=ZX1(IS)+SUMX
   66 CONTINUE
      CONSTX=SUMX/144.
      DO 67 IS=1,8192
      ZZX1(I4)=ZX1(IS)
      I4=I4+1
   67 CONTINUE
      GO TO 37
   36 CONTINUE
      SUMX=0.0
      DO 68 IS=1,144
      SUMX=ZX1(IS)+SUMX
   68 CONTINUE
      AVE1=SUMX/144.
      DO 69 IS=1,8192
      ZZX1(I4)=ZX1(IS)+(CONSTX-AVE1)
      I4=I4+1
   69 CONTINUE
   37 CONTINUE
```

```
      DO 91 I3=1,8192
      ZZY1(I5)=YY(I3)*(1./7.5)
      ZZV1(I5)=ZZ(I3)*10.
      CLFLD(I5)=XXP(I3)
      TIME2(I5)=(DELTAT*FLOAT(I3))+(128.0*FLOAT(K))+XORIGP
      I5=I5+1
   91 CONTINUE
      K=K+1
   70 CONTINUE
C
C     REMOVE DC VALUE OF OUTPUT TIME SERIES DATA IN ORDER TO PLOT
C     FLUCTUATIONS
C
      SUMTT=0.0
      SUMTX=0.0
      DO 223 KJ=1,24576
      SUMTX=SUMTX+CLFLD(KJ)
      SUMTT=SUMTT+ZZX1(KJ)
  223 CONTINUE
      AVG=SUMTT/24576.
      AVX=SUMTX/24576.
      DO 222 JK=1,24576
      CLFLD(JK)=CLFLD(JK)-AVX
      ZZX1(JK)=ZZX1(JK)-AVG
  222 CONTINUE
C
C     THIS NEXT SECTION IS USED TO SMOOTH THE DATA CURVES
C     IT IS NOT NORMALLY USED IN ORDER TO PREVENT BIASING INPUT TIME
C     SERIES DATA, BUT MAY BE USED AT USER'S DISCRETION.  IF USED, NOTE
C     THAT THE PROGRAM OUTPUT WILL NOT BE AS EXACT AS POSSIBLE
C
C     DO 73 L2=1,2
C     Q=0
C     DO 74 IS=1,65318
C     SUMX=0.0
C     SUMY=0.0
C     SUMZ=0.0
C     DO 75 J=1,144
C     SUMX=ZZX1(Q+J)+SUMX
C     SUMY=ZZY1(Q+J)+SUMY
C     SUMZ=ZZV1(Q+J)+SUMZ
C  75 CONTINUE
C     ZZX1(IS)=SUMX/144.
C     ZZY1(IS)=SUMY/144.
C     ZZV1(IS)=SUMZ/144.
C     Q=Q+1
C  74 CONTINUE
C  73 CONTINUE
```

```
C      REAL*8 AH(4),BH(4),CH(4),DH(4),EH(4),FH(4)                            APP11060
C      REAL*8 AL(3),BL(3),CL(3),DL(3),EL(3),FL(3)                            APP11070
C      DIMENSION FRQH(4),FRQL(3)                                             APP11080
C                                                                            APP11090
C      DEFINE AND COMPUTE ALL COEFFICIENTS                                   APP11100
C                                                                            APP11110
C                                                                            APP11120
C      UNDER THE PRESENT DATA COLLECTION SYSTEM, 64 SAMPLES ARE TAKEN        APP11130
C      PER SECOND.  IF ANOTHER DATA COLLECTION SYSTEM IS USED,'T' MUST       APP11140
C      BE ADJUSTED TO THE SAMPLE PERIOD, I.E., 1/SAMPLE RATE                 APP11150
C                                                                            APP11160
       T=1./64.                                                              APP11170
C                                                                            APP11180
C      COEFFICIENTS FOR FIXED HIGH PASS FILTER                               APP11190
C                                                                            APP11200
       AFHP1=-((T**2/160.-2.)/(1.+T/8.+T**2/320.))                           APP11210
       AFHP2=-((1.-T/8.+T**2/320.)/(1.+T/8.+T**2/320.))                      APP11220
       BFHP0=(1./(1.+T/8.+T**2/320.))                                        APP11230
       BFHP1=-(2./(1.+T/8.+T**2/320.))                                       APP11240
       BFHP2=(1./(1.+T/8.+T**2/320.))                                        APP11250
C      WRITE(6,1002)AFHP1,AFHP2,BFHP0,BFHP1,BFHP2                            APP11260
C1002  FORMAT(1X,' FIXED FILTER AFHP1=',F19.16,'AFHP2=',F19.16,'BFHP0=',     APP11270
C     $F19.16,'BFHP1=',F19.16,'BFHP3=',F19.16)                              APP11280
C                                                                            APP11290
C      COEFFICIENTS FOR SELECTABLE HIGH PASS FILTERS                         APP11300
C                                                                            APP11310
C                                                                            APP11320
C      THE FOLLOWING ARRAY VALUES ARE FIXED COEFFICIENTS FOR THE VARIOUS     APP11330
C      FREQUENCY SELECTIONS POSSIBLE ON THE AN/ASQ-81                        APP11340
                                                                             APP11350
       FRQH(1)=0.04                                                          APP11360
       FRQH(2)=0.06                                                          APP11370
       FRQH(3)=0.08                                                          APP11380
       FRQH(4)=0.10                                                          APP11390
       AH(1)=12.52096/40.82834                                               APP11400
       BH(1)=1./40.82834                                                     APP11410
       CH(1)=11.00999/45.28317                                               APP11420
       DH(1)=1./45.28317                                                     APP11430
       EH(1)=7.41498/57.57668                                                APP11440
       FH(1)=1./57.57668                                                     APP11450
       AH(2)=8.34727/18.14591                                                APP11460
       BH(2)=1./18.14591                                                     APP11470
       CH(2)=7.33999/20.12587                                                APP11480
       DH(2)=1./20.12587                                                     APP11490
       EH(2)=4.94332/25.58964                                                APP11500
       FH(2)=1./25.58964                                                     APP11510
       AH(3)=6.26045/10.20708                                                APP11520
       BH(3)=1./10.20708                                                     APP11530
```

```
      CH(3)=5.50500/11.32080
      DH(3)=1./11.32080
      EH(3)=3.70749/14.39417
      FH(3)=1./14.39417
      AH(4)=5.00836/6.53253
      BH(4)=1./6.53253
      CH(4)=4.40400/7.24531
      DH(4)=1./7.24531
      EH(4)=2.96599/9.21227
      FH(4)=1./9.21227
C
C     SELECT THE HIGH PASS FILTER SETTING
C     FOR THE LOW FREQUENCY CUTOFF AT 0.04 HZ, SET I=1
C     FOR THE LOW FREQUENCY CUTOFF AT 0.06 HZ, SET I=2
C     FOR THE LOW FREQUENCY CUTOFF AT 0.08 HZ, SET I=3
C     FOR THE LOW FREQUENCY CUTOFF AT 0.10 HZ, SET I=4
C
      I=1
C
      A1=1.+AH(I)*T/2.+BH(I)*(T**2)/4.
      B1=-2.+BH(I)*(T**2/2.)
      C1=1.-AH(I)*T/2.+BH(I)*(T**2)/4.
      D1=1.+CH(I)*T/2.+DH(I)*(T**2)/4.
      E1=-2.+DH(I)*(T**2)/2.
      F1=1.-CH(I)*T/2.+DH(I)*(T**2)/4.
      G1=1.+EH(I)*T/2.+FH(I)*(T**2)/4.
      H1=-2.+FH(I)*(T**2)/2.
      I1=1.-EH(I)*T/2.+FH(I)*(T**2)/4.
C
C     CODE IS "ASHP1" MEANS "A1 COEFFICIENT FOR THE SELECTABLE HIGH
C     PASS FILTER"
C
      ASHP1=1./(A1*D1*G1)
      ASHP2=-(C1/A1)
      ASHP3=-(B1/A1)
      ASHP4=-(E1/D1)
      ASHP5=-(F1/D1)
      ASHP6=-(H1/G1)
      ASHP7=-(I1/G1)
C     WRITE(6,1000)FRQH(I),ASHP1,ASHP2,ASHP3,ASHP4,ASHP5,ASHP6,ASHP7
C1000 FORMAT(1X,'FREQ=',F5.3,'ASHP1=',F19.16,'ASHP2=',F19.16,'ASHP3=',
C    $F19.16,'ASHP4=',F19.16,/,'ASHP5=',F19.16,'ASHP6=',F19.16,
C    $'ASHP7=',F19.16)
C
C     COEFFIECIENTS FOR SELECTABLE LOW PASS FILTERS
C
      FRQL(1)=0.2
      FRQL(2)=0.4
```

```
      FRQL(3)=0.6
      AL(3)=1./0.C3492
      BL(3)=0.35804/0.03492
      CL(3)=1./0.C3492
      DL(3)=1./0.02779
      EL(3)=0.20696/0.02779
      FL(3)=1./0.02779
      AL(1)=1./0.3143
      BL(1)=1.0741/0.3143
      CL(1)=1./0.3143
      DL(1)=1./0.2501
      EL(1)=0.6209/0.2501
      FL(1)=1./0.2501
      AL(2)=1./0.07858
      BL(2)=0.53706/0.07858
      CL(2)=1./0.07858
      DL(2)=1./0.C6252
      EL(2)=0.31044/0.06252
      FL(2)=1./0.06252
C
C
C     SELECT LOW PASS FILTER SETTING
C     FOR THE HIGH FREQUENCY CUTOFF AT 0.2 HZ, SET J=1
C     FOR THE HIGH FREQUENCY CUTOFF AT 0.4 HZ, SET J=2
C     FOR THE HIGH FREQUENCY CUTOFF AT 0.6 HZ, SET J=3
C
      J=3
C
      A1=AL(J)*(T**2)/4.
      B1=2.*A1
      C1=A1
      D1=DL(J)*(T**2)/4.
      E1=2.*D1
      F1=D1
      G1=(1.+BL(J)*T/2.+CL(J)*(T**2)/4.)
      H1=(-2.+CL(J)*(T**2)/2.)
      I1=(1.-BL(J)*T/2.+CL(J)*(T**2)/4.)
      J1=(1.+EL(J)*T/2.+FL(J)*(T**2)/4.)
      K1=(-2.+FL(J)*(T**2)/2.)
      L1=(1.-EL(J)*T/2.+FL(J)*(T**2)/4.)
      ASLP1=-(G1*K1+H1*J1)/(G1*J1)
      ASLP2=-(G1*L1+H1*K1+I1*J1)/(G1*J1)
      ASLP3=-(H1*L1+I1*K1)/(G1*J1)
      ASLP4=-(I1*L1)/(G1*J1)
      BSLP0=(A1*D1)/(G1*J1)
      BSLP1=(A1*E1+B1*D1)/(G1*J1)
      BSLP2=(A1*F1+B1*E1+C1*D1)/(G1*J1)
      BSLP3=(B1*F1+C1*E1)/(G1*J1)
```

```
      BSLP4=(C1*F1)/(G1*J1)
C     WRITE(6,1001)FRQL(J),ASLP1,ASLP2,ASLP3,ASLP4,BSLP0,BSLP1,BSLP2,
C    $BSLP3,BSLP4
C1001 FORMAT(1X,'FREQ=',F19.16,'ASLP1=',F19.16,'ASLP2=',F19.16,'ASLP3='
C    $,F19.16,'ASLP4=',F19.16,/,'BSLP0=',F19.16,'BSLP1=',F19.16,
C    $'BSLP2=',F19.16,'BSLP3=',F19.16,'BSLP4=',F19.16)
      DO 100 I=1,24576
      SIG=ZZX1(I)
      YO=BFHPO*SIG+BFHP1*SIG1+BFHP2*SIG2+AFHP1*YO1+AFHP2*YO2
      XI=ASHP1*YO+ASHP2*XI2+ASHP3*XI1
      XII=XI+XI2-2.*XI1
      XIII=XII+ASHP4*XIII1+ASHP5*XIII2
      XIV=XIII-2.*XIII1+XIII2
      XV=XIV+ASHP6*XV1+ASHP7*XV2
      YPO=XV+XV2-2.*XV1
      GP1=ASLP1*OUTFD1+ASLP2*OUTFD2+ASLP3*OUTFD3+ASLP4*OUTFD4
      GP2=BSLP0*YPO+BSLP1*YPO1+BSLP2*YPO2+BSLP3*YPO3+BSL4*YPO4
      OUTFLD(I)=GP1+GP2
C
C     FINISHED COMPUTING THIS STEP'S VALUES FOR AMPLITUDES
C     INCREMENT STORAGE REGISTERS
C
      SIG2=SIG1
      SIG1=SIG
      YO2=YO1
      YO1=YO
      XI2=XI1
      XI1=XI
      XIII2=XIII1
      XIII1=XIII
      XV2=XV1
      XV1=XV
      YPO4=YPO3
      YPO3=YPO2
      YPO2=YPO1
      YPO1=YPO
      OUTFD4=OUTFD3
      OUTFD3=OUTFD2
      OUTFD2=OUTFD1
      OUTFD1=OUTFLD(I)
C
C     FINISHED INCREMENTING STORAGE REGISTERS
C
  100 CONTINUE
      XMAXP=TIME2(16384)
C
C    VERSATEC PLOT OF B - FIELD SPECTRA
C
```

```
C      NPTS=1020./CELTAT +1.
       NPTS=24576
C        'NPTS' DETERMINES NUMBER OF POINTS NECESSARY IN ORDER FOR
C         THE 0 TO 2041  SECS RANGE TO BE PLOTTED.
C        FOR THE FOLLOWING 'ITB' AND 'RTB' VALUES REVIEW THE WRITE-UP
C        FOR THE SUBROUTINE PROCEDURE 'DRAWP'.
       ITB(3)=8
       ITB(4)=4
       ITB(7)=1
       ITB(12)=0
       RTB(1)=0.0
       RTB(2)=0.0
       RTB(3)=ALAB(1)
       READ(5,3000)TITLE
C
C      DRAW THE COIL ANTENNA TOTAL FIELD DATA  SERIES
C
       CALL DRAWP(NPTS,TIME2,ZZX1,ITB,RTB)
       RTB(3)=ALAB(2)
       READ(5,3000)TITLE
C
C      DRAW THE ASQ81  TOTAL FIELD DATA SERIES
C
       CALL DRAWP(NPTS,TIME2,ZZY1,ITB,RTB)
       RTB(3)=ALAB(3)
       READ(5,3000)TITLE
C
C      DRAW THE SCHONSTEDT COIL FIELD DATA SERIES
C
       CALL DRAWP(NPTS,TIME2,ZZV1,ITB,RTB)
       RTB(3)=ALAB(4)
       READ(5,3000)TITLE
C
C      DRAW THE PROGRAM OUTPUT TOTAL FIELD DATA SERIES
C
       CALL DRAWP(NPTS,TIME2,OUTFLD,ITB,RTB)
C
C      DRAW THE RAW COIL TIME SERIES DATA
C
       RTB(3)=ALAB(4)
       READ(5,3000)TITLE
       CALL DRAWP(NPTS,TIME2,CLFLD,ITB,RTB)
  200  CONTINUE
 3000  FORMAT(6A8)
       STOP
       END
        SUBROUTINE RD(IUN,IO,IRS,IREC,IRQ)
C
```

```
C           THIS PRCCEDURE FURNISHED BY DR. TIM STANTON,
C           DEPARTMENT OF OCEANOGRAPHY.
C
C                READ DATA FROM IUN, ALLIGN , CHECK & RETJRN
C
C         IUN=TAPE NUMBER, EG 20
C         IO=INTEGER*2 ARRAY, 16 LONG,(VALUES 0-4095, SUBTRACT 2048)*5
C                 /2028.  GIVES VOLTAGE
C         IRS= NUMBER OF RESINCS ALLOWED (ERRORS)
C         IREC= COUNTER OF RECORDS (FRAMES OF DATA)
C             BLOCK 512 BITS,   32 BITS = RECORD
C             800 BPI TAPE UNLABLED
C         IRQ= NUMBER OF ACTUAL RESINCS (ERRORS)
C
C
        INTEGER * 2 IO(16),IP(16)
        DATA IRR /0/
        IF (IREC.EQ.0) IS=0
        IER=0
 20     FORMAT (16A2)
        IF (IS.NE.0) GO TO 50
        READ (IUN,20,END=900) IP
        IREC=IREC+1
 40     IS=IS+1
        IF (IS.LT.17) GO TO 50
        READ (IUN,20,END=900) IP
       IS=1
       IREC=IREC+1
 50    ICH=IMASK(IP(IS),3,0)+1
C       WRITE (6,55) ICH,IS,IUN,IREC
 55     FORMAT ('  RESYNCING ICH,IS,IUN,IREC ',4I8)
C
       IF (ICH.NE.1) GO TO 40
       DO 100 I=1,16
        IO(I)=ISHIFT(IP(IS),4)
       ICH=IMASK(IP(IS),3,0)+1
       IF (ICH.EQ.1) GO TO 80
       IER=IER+1
       WRITE (6,70)  IUN,IREC ,I,ICH,IER
 70     FORMAT (' UNIT ',I3,' RECORD ',I6,'CHAN & DATA CH ',2I4,
     $ 'ERRORS ',I7)
 80     IS=IS+1
        IF (IS.LT.17) GO TO 100
        READ (IUN,20,END=900) IP
        IS=1
        IREC=IREC+1
 100    CONTINUE
C
```

```
      IF (IER.EQ.0) GO TO 150                                           APP13940
      IRR=IRR+1                                                         APP13950
      IF (IRR.LT.IRS) GO TO 120                                         APP13960
      WRITE (6,110)                                                     APP13970
110   FORMAT ('1 STOPPED IN SUB RD BECAUSE OF IRR.GT.',I6,' AT L110')   APP13980
      IRQ=IRR                                                           APP13990
      STOP                                                              APP14000
120   CONTINUE                                                          APP14010
      WRITE (6,130) IREC,IRR                                            APP14020
130   FORMAT (' RESYNC AT FRAME ',I6,' WITH TOTAL ERRORS ',I7)          APP14030
      IER=0                                                             APP14040
      IRQ=IRR                                                           APP14050
      GO TO 50                                                          APP14060
150   CONTINUE                                                          APP14070
      RETURN                                                            APP14080
900   WRITE (6,910) IUN,IREC                                            APP14090
910   FORMAT ('1   END OF UNIT ',I3,' AT REC ',I7)                      APP14100
      STOP                                                              APP14110
      END                                                               APP14120
                                                                        APP14130
      FUNCTION ISHIFT (IN,NPLC)                                         APP14140
C              RETURNS SHIFTED VALUE OF I*2 WORD IN                     APP14150
C              -VE LEFT,+VE RIGHT SHIFT                                 APP14160
C                                                                       APP14170
      INTEGER * 2 IN                                                    APP14180
      IP=IN                                                             APP14190
      IF (IP.LT.0) IP=IP+65536                                          APP14200
      IF (NPLC.LT.0) GO TO 30                                           APP14210
      ISHIFT=IP/(2**IABS(NPLC))                                         APP14220
      RETURN                                                            APP14230
30    ISHIFT=IP*(2**IABS(NPLC))                                         APP14240
      IF (ISHIFT.GT.65535) ISHIFT=MOD(ISHIFT,65536)                     APP14250
      RETURN                                                            APP14260
      END                                                               APP14270
      FUNCTION IMASK (IN,IBL,IBR)                                       APP14280
C              MASK I*2 WORD IN OUTSIDE BITS IBL & IBR                  APP14290
C                                                                       APP14300
      INTEGER * 2 IN,IO                                                 APP14310
      IO=IN                                                             APP14320
      IF (IBR.EQ.0) GO TO 50                                            APP14330
      IT=ISHIFT(IN,IBR)                                                 APP14340
      IO=IT                                                             APP14350
50    IP=ISHIFT(IO,IBL-15-IBR)                                          APP14360
      IO=IP                                                             APP14370
      IMASK=ISHIFT(IO,15-IBL)                                           APP14380
      RETURN                                                            APP14390
      END                                                               APP14400
C     SUBROUTINE FOURT                                                  APP14410
```

```
C
C     PURPOSE
C
C     SUBROUTINE FOURT COMPUTES THE FORWARD AND INVERSE
C     COOLEY-TUKEY FAST FOURIER TRANSFORM OF THE CONTENTS OF THE
C     ARRAY DATA.  FOR DATA A SINGLY-DIMENSIONED ARRAY OF LENGTH
C     L, THE JTH COMPONENT OF THE TRANSFORM IS GIVEN BY
C        SUM(DATA(K)*W**(K-1)*(J-1))
C     WHERE THE SUM IS TAKEN OVER K, 1 .LE. K .LE. L, AND
C        W=EXP(ISIGN*2*PI*SQRT(-1)/L)
C
C     THE VALUE OF ISIGN DEPENDS UPON WHETHER A FORWARD OR INVERSE
C     TRANSFORM IS TO BE PERFORMED.  FOURT MAY ALSO BE USED ON A
C     MULTI-DIMENSIONAL ARRAY, IN WHICH CASE A FOURIER TRANSFORM IS
C     PERFORMED ALONG EACH DIMENSION IN TURN.
C
C     CALLING SEQUENCE
C
C     CALL FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)
C
C     DESCRIPTION OF ARGUMENTS
C
C     DATA   COMPLEX*8 MULTI-DIMENSIONAL ARRAY CONTAINING THE DATA TO
C            BE TRANSFORMED.  ON OUTPUT DATA CONTAINS THE TRANSFORM.
C            NORMAL FORTRAN ORDERING IS EXPECTED, THE FIRST SUBSCRIPT
C            CHANGING THE FASTEST.
C
C     NN     INTEGER*4 ARRAY CONTAINING THE DIMENSIONS OF THE ARRAY
C            DATA.
C
C     NDIM   NUMBER OF DIMENSIONS OF THE ARRAY DATA = NUMBER OF ELE-
C            MENTS IN THE ARRAY NN.
C
C     ISIGN  INTEGER INDICATING WHETHER FORWARD OR INVERSE TRANSFORM
C            IS TO BE PERFORMED.
C              ISIGN=-1 FOR FORWARD TRANSFORM
C              ISIGN=1 FOR INVERSE TRANSFORM.
C            NOTE:  THESE DEFINITIONS ARE NOT STANDARDIZED.  IN PAR-
C                   TICULAR, THE DEFINITIONS OF FORWARD AND INVERSE
C                   TRANSFORM ARE REVERSED IN THE IMSL FFT ROUTINES.
C
C     IFORM  AN INTEGER INDICATING WHETHER OR NOT DATA CONTAINS ONLY
C            PURELY REAL VALUES.
C              IFORM=0 IF DATA IS PURELY REAL
C              IFORM=1 OTHERWISE.
C            IF IFORM IS SET TO 0, ALL THE IMAGINARY PARTS OF THE ELE-
C            MENTS IN DATA MUST BE SET TO 0.0.
C
```

```
C     WORK   A 1-DIMENSIONAL REAL*4 ARRAY USED FOR WORKING STORAGE.
C            ITS LENGTH SHOULD BE TWICE THE LARGEST ARRAY DIMENSION
C            NN(I), I=1,2,...NDIM, WHICH IS NOT A POWER OF TWO.  IN
C            PARTICULAR, IF ALL NN(I) ARE POWERS OF TWO, NO WORK SPACE
C            IS NEEDED AND WORK MAY BE REPLACED BY ZERO IN THE CALLING
C            SEQUENCE.
C
C     REMARKS
C
C     IF AN INVERSE TRANSFORM (ISIGN=+1) IS PERFORMED UPON AN ARRAY
C     OF TRANSFORMED (ISIGN=-1) DATA, THE ORIGINAL DATA WILL REAP-
C     PEAR, MULTIPLIED BY NN(1)*NN(2)*...*NN(NDIM).
C
C     FOR A MULTI-DIMENSIONAL ARRAY THE (J1,J2,...,JNDIM)
C     COMPONENT OF THE TRANSFORM IS GIVEN BY
C        SUM(DATA(I1,I2,...INDIM)*W1**((I1-1)*(J1-1))*
C        W2**((I2-1)*(J2-1)*...*WNDIM**((INDIM-1)*(JNDIM-1))
C     HERE THE SUM RANGES OVER ALL POSSIBLE VALUES OF THE I'S
C     AND W1=EXP(ISIGN*2*PI*SQRT(-1)/NN(1)), ETC.
C
C     THE ARRAY OF INPUT DATA MUST BE IN COMPLEX FORMAT.
C     HOWEVER, IS ALL IMAGINARY PARTS ARE ZERO (I.E., THE DATA
C     ARE DISGUISED REAL) RUNNING TIME IS CUT UP TO FORTY PER-
C     CENT.  (FOR FASTEST TRANSFORM OF REAL DATA, NN(1) SHOULD BE E-
C     VEN.)  THE TRANSFORM VALUES ARE ALWAYS COMPLEX AND ARE RETURNED
C     IN THE ORIGINAL ARRAY OF DATA, REPLACING THE INPUT DATA.  THE
C     LENGTH OF EACH DIMENSION OF THE DATA ARRAY MAY BE ANY INTEGER.
C     THE PROGRAM RUNS FASTER ON COMPOSITE INTEGERS THAN ON PRIMES,
C     AND IS PARTICULARLY FAST ON NUMBERS RICH IN FACTORS OF TWO.
C
C     TIMING IS IN FACT GIVEN BY THE FOLLOWING FORMULA.  LET NTOT BE
C     THE TOTAL NUMBER OF POINTS (REAL OR COMPLEX) IN THE DATA ARRAY,
C     THAT IS, NTOT=NN(1)*NN(2)*...  DECOMPOSE NTOT INTO ITS PRIME
C     FACTORS, SUCH AS 2**K2 * 3**K3 * 5**K5 * ...  LET SUM2 BE THE
C     SUM OF THE FACTORS OF TWO IN NTOT, THAT IS, SUM2 = 2*K2.  LET
C     SUMF BE THE SUM OF ALL OTHER FACTORS OF NTOT, THAT IS, SUMF =
C     3*K3*5*K5*..  THE TIME TAKEN BY A MULTIDIMENSIONAL TRANSFORM ON
C     THESE NTOT DATA IS T = T0 + NTOT*(T1+T2*SUM2+T3*SUMF).  ON THE
C     CDC 3300 (FLOATING POINT ADD TIME = SIX MICROSECONDS), T = 3000+
C     NTOT*(600+40*SUM2+175*SUMF) MICROSECONDS ON COMPLEX DATA.
C
C     THE SAVINGS OFFERED BY THIS PROGRAM CAN BE DRAMATIC.  A ONE-DI-
C     MENSIONAL ARRAY 4000 IN LENGTH WILL BE TRANSFORMED IN 4000*(600+
C     40*(2+2+2+2+2)+175*(5+5+5)) = 14.5 SECONDS VERSUS ABOUT 4000*
C     4000*175 = 2800 SECONDS FOR THE STRAIGHTFORWARD TECHNIQUE.
C
C     THE FAST FOURIER TRANSFORM PLACES THREE RESTRICTIONS UPON THE
C     DATA.
```

```
C       1.  THE NUMBER OF INPUT DATA AND THE NUMBER OF TRANSFORM VALUES    APP15380
C           MUST BE THE SAME.                                              APP15390
C       2.  BOTH THE INPUT DATA AND THE TRANSFORM VALUES MUST REPRESENT    APP15400
C           EQUISPACED POINTS IN THEIR RESPECTIVE DOMAINS OF TIME AND      APP15410
C           FREQUENCY.  CALLING THESE SPACINGS DELTAT AND DELTAF, IT       APP15420
C           MUST BE TRUE THAT DELTAF=2*PI/(NN(I)*DELTAT).  OF COURSE,     APP15430
C           DELTAT NEED NOT BE THE SAME FOR EVERY DIMENSION.               APP15440
C       3.  CONCEPTUALLY AT LEAST, THE INPUT DATA AND THE TRANSFORM OUT-   APP15450
C           PUT REPRESENT SINGLE CYCLES OF PERIODIC FUNCTIONS.             APP15460
C                                                                          APP15470
C       THERE ARE NC ERROR MESSAGES OR ERROR HALTS IN THIS PROGRAM.  THE   APP15480
C       PROGRAM RETURNS IMMEDIATELY IF NDIM OR ANY NN(I) IS LESS THAN      APP15490
C       ONE.                                                               APP15500
C                                                                          APP15510
C       FOR MOST APPLICATIONS FOURT, IF COMPILED UNDER FORTRAN H, IS       APP15520
C       COMPARABLE IN SPEED AND ACCURACY TO THE IMSL FFT SUBROUTINES.      APP15530
C       WITH CERTAIN PATHOLOGICALLY ILL-CONDITIONED DATA THE               APP15540
C       ACCURACY OF FOURT MAY BE SERIOUSLY DEGRADED, BUT THE               APP15550
C       SAME CAN PROBABLY BE SAID OF ANY EXTANT FFT ROUTINE.               APP15560
C       WORK SPACE REQUIRED BY FOURT MAY BE GREATER OR LESS THAN THAT      APP15570
C       REQUIRED BY THE IMSL ROUTINES, DEPENDING UPON THE APPLICATION.     APP15580
C       FOURT IS MORE FLEXIBLE AND IN GENERAL EASIER TO USE THAN THE       APP15590
C       IMSL ROUTINES.  FOURT ALONE PROVIDES THE CAPABILITY OF TRANS-      APP15600
C       FORMING A MULTI-DIMENSIONAL ARRAY WITH A SINGLE CALL.              APP15610
C                                                                          APP15620
C       THIS IS THE FASTEST AND MOST VERSATILE VERSION OF THE FFT KNOWN    APP15630
C       TO THE AUTHOR.  A PROGRAM CALLED FOUR2 IS AVAILABLE THAT ALSO      APP15640
C       PERFORMS THE FAST FOURIER TRANSFORM AND IS WRITTEN IN USASI BA-    APP15650
C       SIC FORTRAN.  IT IS ABOUT ONE THIRD AS LONG AND RESTRICTS THE      APP15660
C       DIMENSICNS OF THE INPUT ARRAY (WHICH MUST BE COMPLEX) TO BE        APP15670
C       POWERS OF TWO.  ANOTHER PROGRAM, CALLED FOUR1, IS ONE TENTH AS     APP15680
C       LONG AND RUNS TWO THIRDS AS FAST ON A ONE-DIMENSIONAL COMPLEX      APP15690
C       ARRAY WHOSE LENGTH IS A POWER OF TWO.                              APP15700
C                                                                          APP15710
C       REFERENCE--                                                        APP15720
C       IEEE AUDIO TRANSACTIONS (JUNE 1967), SPECIAL ISSUE ON THE FFT.     APP15730
C                                                                          APP15740
C       EXAMPLE 1.  THREE-DIMENSIONAL FORWARD FOURIER TRANSFORM OF A       APP15750
C       COMPLEX ARRAY DIMENSIONED 32 BY 25 BY 13 IN FORTRAN IV.            APP15760
C       DIMENSION DATA(32,25,13),WORK(50),NN(3)                            APP15770
C       COMPLEX DATA                                                       APP15780
C       DATA NN/32,25,13/                                                  APP15790
C       DO 1 I=1,32                                                        APP15800
C       DO 1 J=1,25                                                        APP15810
C       DO 1 K=1,13                                                        APP15820
C     1 DATA(I,J,K)=COMPLEX VALUE                                          APP15830
C       CALL FOURT(DATA,NN,3,-1,1,WORK)                                    APP15840
C                                                                          APP15850
```

```
C       EXAMPLE 2.  ONE-DIMENSIONAL FORWARD TRANSFORM OF A REAL ARRAY OF   APP15860
C       LENGTH 64 IN FORTRAN II                                            APP15870
C       DIMENSION DATA(2,64)                                               APP15880
C       DO 2 I=1,64                                                        APP15890
C       DATA(1,I)=REAL PART                                                APP15900
C   2   DATA(2,I)=0.                                                       APP15910
C       CALL FOURT(DATA,64,1,-1,0,0)                                       APP15920
C                                                                          APP15930
C       PROGRAMMER                                                         APP15940
C                                                                          APP15950
C       PROGRAM BY NORMAN BRENNER FROM THE BASIC PROGRAM BY CHARLES        APP15960
C       RADER, JUNE 1967.  THE IDEA FOR THE DIGIT REVERSAL WAS             APP15970
C       SUGGESTED BY RALPH ALTER.                                          APP15980
C       DOCUMENTATION REVISED BY JOANNE BOGART, AUGUST 1979, NPS.          APP15990
C                                                                          APP16000
        SUBROUTINE FOURT(DATA,NN,NDIM,ISIGN,IFORM,WORK)                    APP16010
        DIMENSION DATA(1),NN(1),IFACT(32),WORK(1)                          APP16020
        DATA TWOPI/6.283185307179 6/,RTHLF/0.70710678118655/               APP16030
        IF(NDIM-1)920,1,1                                                  APP16040
  1     NTOT=2                                                             APP16050
        DO 2 IDIM=1,NDIM                                                   APP16060
        IF(NN(IDIM))920,920,2                                              APP16070
  2     NTOT=NTOT*NN(IDIM)                                                 APP16080
C                                                                          APP16090
C       MAIN LOOP FOR EACH DIMENSION                                       APP16100
C                                                                          APP16110
        NP1=2                                                              APP16120
        DO 910 IDIM=1,NDIM                                                 APP16130
        N=NN(IDIM)                                                         APP16140
        NP2=NP1*N                                                          APP16150
        IF(N-1)920,900,5                                                   APP16160
C                                                                          APP16170
C       IS N A POWER OF TWO AND IF NOT, WHAT ARE ITS FACTORS               APP16180
C                                                                          APP16190
  5     M=N                                                                APP16200
        NTWO=NP1                                                           APP16210
        IF=1                                                               APP16220
        IDIV=2                                                             APP16230
 10     IQUOT=M/IDIV                                                       APP16240
        IREM=M-IDIV*IQUOT                                                  APP16250
        IF(IQUOT-IDIV)50,11,11                                             APP16260
 11     IF(IREM)20,12,20                                                   APP16270
 12     NTWO=NTWO+NTWO                                                     APP16280
        IFACT(IF)=IDIV                                                     APP16290
        IF=IF+1                                                            APP16300
        M=IQUOT                                                            APP16310
        GO TO 10                                                           APP16320
 20     IDIV=3                                                             APP16330
```

```
      INON2=IF                                                          APP16340
   30 IQUOT=M/IDIV                                                      APP16350
      IREM=M-IDIV*IQUOT                                                 APP16360
      IF(IQUOT-IDIV)60,31,31                                            APP16370
   31 IF(IREM)40,32,40                                                  APP16380
   32 IFACT(IF)=IDIV                                                    APP16390
      IF=IF+1                                                           APP16400
      M=IQUOT                                                           APP16410
      GO TO 30                                                          APP16420
   40 IDIV=IDIV+2                                                       APP16430
      GO TO 30                                                          APP16440
   50 INON2=IF                                                          APP16450
      IF(IREM)60,51,60                                                  APP16460
   51 NTWO=NTWO+NTWO                                                    APP16470
      GO TO 70                                                          APP16480
   60 IFACT(IF)=M                                                       APP16490
C                                                                       APP16500
C     SEPARATE FOUR CASES--                                             APP16510
C        1. COMPLEX TRANSFORM OR REAL TRANSFORM FOR THE 4TH, 9TH, ETC.  APP16520
C           DIMENSIONS.                                                 APP16530
C        2. REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION.  METHOD--      APP16540
C           TRANSFORM HALF THE DATA, SUPPLYING THE OTHER HALF BY CON-   APP16550
C           JUGATE SYMMETRY.                                            APP16560
C        3. REAL TRANSFORM FOR THE 1ST DIMENSION, N ODD.  METHOD--      APP16570
C           SET THE IMAGINARY PARTS TO ZERO.                            APP16580
C        4. REAL TRANSFORM FOR THE 1ST DIMENSION, N EVEN.  METHOD--     APP16590
C           TRANSFORM A COMPLEX ARRAY OF LENGTH N/2 WHOSE REAL PARTS    APP16600
C           ARE THE EVEN NUMBERED REAL VALUES AND WHOSE IMAGINARY       APP16610
C           PARTS ARE THE ODD NUMBERED REAL VALUES.  SEPARATE AND SUP-  APP16620
C           PLY THE SECOND HALF BY CONJUGATE SYMMETRY.                  APP16630
C                                                                       APP16640
   70 ICASE=1                                                           APP16650
      IFMIN=1                                                           APP16660
      I1RNG=NP1                                                         APP16670
      IF(IDIM-4)71,100,100                                              APP16680
   71 IF(IFORM)72,72,100                                                APP16690
   72 ICASE=2                                                           APP16700
      I1RNG=NP0*(1+NPREV/2)                                             APP16710
      IF(IDIM-1)73,73,100                                               APP16720
   73 ICASE=3                                                           APP16730
      I1RNG=NP1                                                         APP16740
      IF(NTWO-NP1)100,100,74                                            APP16750
   74 ICASE=4                                                           APP16760
      IFMIN=2                                                           APP16770
      NTWO=NTWO/2                                                       APP16780
      N=N/2                                                             APP16790
      NP2=NP2/2                                                         APP16800
      NTOT=NTOT/2                                                       APP16810
```

```
      I=1                                                               APP16820
      DO 80  J=1,NTOT                                                   APP16830
      DATA(J)=DATA(I)                                                   APP16840
   80 I=I+2                                                             APP16850
C                                                                       APP16860
C     SHUFFLE DATA BY BIT REVERSAL, SINCE N=2**K.  AS THE SHUFFLING     APP16870
C     CAN BE DONE BY SIMPLE INTERCHANGE, NO WORKING ARRAY IS NEEDED     APP16880
C                                                                       APP16890
  100 IF(NTWO-NP2)200,110,110                                           APP16900
  110 NP2HF=NP2/2                                                       APP16910
      J=1                                                               APP16920
      DO 150  I2=1,NP2,NP1                                              APP16930
      IF(J-I2)120,130,130                                               APP16940
  120 I1MAX=I2+NP1-2                                                    APP16950
      DO 125  I1=I2,I1MAX,2                                             APP16960
      DO 125  I3=I1,NTOT,NP2                                            APP16970
      J3=J+I3-I2                                                        APP16980
      TEMPR=DATA(I3)                                                    APP16990
      TEMPI=DATA(I3+1)                                                  APP17000
      DATA(I3)=DATA(J3)                                                 APP17010
      DATA(I3+1)=DATA(J3+1)                                             APP17020
      DATA(J3)=TEMPR                                                    APP17030
  125 DATA(J3+1)=TEMPI                                                  APP17040
  130 M=NP2HF                                                           APP17050
  140 IF(J-M)150,150,145                                                APP17060
  145 J=J-M                                                             APP17070
      M=M/2                                                             APP17080
      IF(M-NP1)150,140,140                                              APP17090
  150 J=J+M                                                             APP17100
      GO TO 300                                                         APP17110
C                                                                       APP17120
C     SHUFFLE DATA BY DIGIT REVERSAL FOR GENERAL NG ARRAY IS NEEDED     APP17130
C                                                                       APP17140
  200 NWORK=2*N                                                         APP17150
      DO 270  I1=1,NP1,2                                                APP17160
      DO 270  I3=I1,NTOT,NP2                                            APP17170
      J=I3                                                              APP17180
      DO 260  I=1,NWORK,2                                               APP17190
      IF(ICASE-3)210,220,210                                            APP17200
  210 WORK(I)=DATA(J)                                                   APP17210
      WORK(I+1)=DATA(J+1)                                               APP17220
      GO TO 230                                                         APP17230
  220 WORK(I)=DATA(J)                                                   APP17240
      WORK(I+1)=0.                                                      APP17250
  230 IFP2=NP2                                                          APP17260
      IF=IFMIN                                                          APP17270
  240 IFP1=IFP2/IFACT(IF)                                               APP17280
      J=J+IFP1                                                          APP17290
```

```
      IF(J-I3-IFP2)260,250,250                                          APP17300
  250 J=J-IFP2                                                          APP17310
      IFP2=IFP1                                                         APP17320
      IF=IF+1                                                           APP17330
      IF(IFP2-NP1)260,260,240                                           APP17340
  260 CONTINUE                                                          APP17350
      I2MAX=I3+NP2-NP1                                                  APP17360
      I=1                                                               APP17370
      DO 270 I2=I3,I2MAX,NP1                                            APP17380
      DATA(I2)=WORK(I)                                                  APP17390
      DATA(I2+1)=WORK(I+1)                                              APP17400
  270 I=I+2                                                             APP17410
C                                                                       APP17420
C     MAIN LOOP FOR FACTORS OF TWO.  PERFORM FOURIER TRANSFORMS OF      APP17430
C     LENGTH FOUR, WITH ONE OF LENGTH TWO IF NEEDED.  THE TWIDDLE FAC-  APP17440
C     TOR W=EXP(ISIGN*2*PI*SQRT(-1)*M/(4*MMAX)).  CHECK FOR W=ISIGN*    APP17450
C     SQRT(-1)                                                          APP17460
C                                                                       APP17470
  300 IF(NTWO-NP1)600,600,305                                           APP17480
  305 NP1TW=NP1+NP1                                                     APP17490
      IPAR=NTWO/NP1                                                     APP17500
  310 IF(IPAR-2)350,330,320                                             APP17510
  320 IPAR=IPAR/4                                                       APP17520
      GO TO 310                                                         APP17530
  330 DO 340 I1=1,I1RNG,2                                               APP17540
      DO 340 K1=I1,NTOT,NP1TW                                           APP17550
      K2=K1+NP1                                                         APP17560
      TEMPR=DATA(K2)                                                    APP17570
      TEMPI=DATA(K2+1)                                                  APP17580
      DATA(K2)=DATA(K1)-TEMPR                                           APP17590
      DATA(K2+1)=DATA(K1+1)-TEMPI                                       APP17600
      DATA(K1)=DATA(K1)+TEMPR                                           APP17610
  340 DATA(K1+1)=DATA(K1+1)+TEMPI                                       APP17620
  350 MMAX=NP1                                                          APP17630
  360 IF(MMAX-NTWO/2)370,600,600                                        APP17640
  370 LMAX=MAX0(NP1TW,MMAX/2)                                           APP17650
      DO 570 L=NP1,LMAX,NP1TW                                           APP17660
      M=L                                                               APP17670
      IF(MMAX-NP1)420,420,380                                           APP17680
  380 THETA=-TWOPI*FLOAT(L)/FLOAT(4*MMAX)                               APP17690
      IF(ISIGN)400,390,390                                              APP17700
  390 THETA=-THETA                                                      APP17710
  400 WR=COS(THETA)                                                     APP17720
      WI=SIN(THETA)                                                     APP17730
  410 W2R=WR*WR-WI*WI                                                   APP17740
      W2I=2.*WR*WI                                                      APP17750
      W3R=W2R*WR-W2I*WI                                                 APP17760
      W3I=W2R*WI+W2I*WR                                                 APP17770
```

```
420   DO 530 I1=1,I1RNG,2                                        APP17780
      KMIN=I1+IPAR*M                                             APP17790
      IF(MMAX-NP1)430,430,440                                    APP17800
430   KMIN=I1                                                    APP17810
440   KDIF=IPAR*MMAX                                             APP17820
450   KSTEP=4*KDIF                                               APP17830
      IF(KSTEP-NTWO)460,460,530                                  APP17840
460   DO 520 K1=KMIN,NTOT,KSTEP                                  APP17850
      K2=K1+KDIF                                                 APP17860
      K3=K2+KDIF                                                 APP17870
      K4=K3+KDIF                                                 APP17880
      IF(MMAX-NP1)470,470,480                                    APP17890
470   U1R=DATA(K1)+DATA(K2)                                      APP17900
      U1I=DATA(K1+1)+DATA(K2+1)                                  APP17910
      U2R=DATA(K3)+DATA(K4)                                      APP17920
      U2I=DATA(K3+1)+DATA(K4+1)                                  APP17930
      U3R=DATA(K1)-DATA(K2)                                      APP17940
      U3I=DATA(K1+1)-DATA(K2+1)                                  APP17950
      IF(ISIGN)471,472,472                                       APP17960
471   U4R=DATA(K3+1)-DATA(K4+1)                                  APP17970
      U4I=DATA(K4)-DATA(K3)                                      APP17980
      GO TO 510                                                  APP17990
472   U4R=DATA(K4+1)-DATA(K3+1)                                  APP18000
      U4I=DATA(K3)-DATA(K4)                                      APP18010
      GO TO 510                                                  APP18020
480   T2R=W2R*DATA(K2)-W2I*DATA(K2+1)                            APP18030
      T2I=W2R*DATA(K2+1)+W2I*DATA(K2)                            APP18040
      T3R=WR*DATA(K3)-WI*DATA(K3+1)                              APP18050
      T3I=WR*DATA(K3+1)+WI*DATA(K3)                              APP18060
      T4R=W3R*DATA(K4)-W3I*DATA(K4+1)                            APP18070
      T4I=W3R*DATA(K4+1)+W3I*DATA(K4)                            APP18080
      U1R=DATA(K1)+T2R                                           APP18090
      U1I=DATA(K1+1)+T2I                                         APP18100
      U2R=T3R+T4R                                                APP18110
      U2I=T3I+T4I                                                APP18120
      U3R=DATA(K1)-T2R                                           APP18130
      U3I=DATA(K1+1)-T2I                                         APP18140
      IF(ISIGN)490,500,500                                       APP18150
490   U4R=T3I-T4I                                                APP18160
      U4I=T4R-T3R                                                APP18170
      GO TO 510                                                  APP18180
500   U4R=T4I-T3I                                                APP18190
      U4I=T3R-T4R                                                APP18200
510   DATA(K1)=U1R+U2R                                           APP18210
      DATA(K1+1)=U1I+U2I                                         APP18220
      DATA(K2)=U3R+U4R                                           APP18230
      DATA(K2+1)=U3I+U4I                                         APP18240
      DATA(K3)=U1R-U2R                                           APP18250
```

```
      DATA(K3+1)=U1I-U2I                                                APP18260
      DATA(K4)=U3R-U4R                                                  APP18270
  520 DATA(K4+1)=U3I-U4I                                                APP18280
      KDIF=KSTEP                                                        APP18290
      KMIN=4*(KMIN-I1)+I1                                               APP18300
      GO TO 450                                                         APP18310
  530 CONTINUE                                                          APP18320
      M=M+LMAX                                                          APP18330
      IF(M-MMAX)540,540,570                                             APP18340
  540 IF(ISIGN)550,560,560                                              APP18350
  550 TEMPR=WR                                                          APP18360
      WR=(WR+WI)*RTHLF                                                  APP18370
      WI=(WI-TEMPR)*RTHLF                                               APP18380
      GO TO 410                                                         APP18390
  560 TEMPR=WR                                                          APP18400
      WR=(WR-WI)*RTHLF                                                  APP18410
      WI=(TEMPR+WI)*RTHLF                                               APP18420
      GO TO 410                                                         APP18430
  570 CONTINUE                                                          APP18440
      IPAR=3-IPAR                                                       APP18450
      MMAX=MMAX+MMAX                                                    APP18460
      GO TO 360                                                         APP18470
C                                                                       APP18480
C     MAIN LOOP FOR FACTORS NOT EQUAL TO TWO.  APPLY THE TWIDDLE FAC-   APP18490
C     TOR W=EXP(ISIGN*2*PI*SQRT(-1)*(J1-1)*(J2-J1)/(IFP1+IFP2)),        APP18500
C     THEN PERFORM A FOURIER TRANSFORM OF LENGTH IFACT(IF), MAKING USE  APP18510
C     OF CONJUGATE SYMMETRIES.                                          APP18520
C                                                                       APP18530
  600 IF(NTWO-NP2)605,700,700                                           APP18540
  605 IFP1=NTWO                                                         APP18550
      IF=INON2                                                          APP18560
      NP1HF=NP1/2                                                       APP18570
  610 IFP2=IFACT(IF)*IFP1                                               APP18580
      J1MIN=NP1+1                                                       APP18590
      IF(J1MIN-IFP1)615,615,640                                         APP18600
  615 DO 635 J1=J1MIN,IFP1,NP1                                          APP18610
      THETA=-TWOPI*FLOAT(J1-1)/FLOAT(IFP2)                              APP18620
      IF(ISIGN)625,620,620                                              APP18630
  620 THETA=-THETA                                                      APP18640
  625 WSTPR=COS(THETA)                                                  APP18650
      WSTPI=SIN(THETA)                                                  APP18660
      WR=WSTPR                                                          APP18670
      WI=WSTPI                                                          APP18680
      J2MIN=J1+IFP1                                                     APP18690
      J2MAX=J1+IFP2-IFP1                                                APP18700
      DO 635 J2=J2MIN,J2MAX,IFP1                                        APP18710
      I1MAX=J2+I1RNG-2                                                  APP18720
      DO 630 I1=J2,I1MAX,2                                              APP18730
```

```
      DO 630 J3=I1,NTOT,IFP2                                            APP18740
      TEMPR=DATA(J3)                                                    APP18750
      DATA(J3)=DATA(J3)*WR-DATA(J3+1)*WI                                APP18760
  630 DATA(J3+1)=TEMPR*WI+DATA(J3+1)*WR                                 APP18770
      TEMPR=WR                                                          APP18780
      WR=WR*WSTPR-WI*WSTPI                                              APP18790
  635 WI=TEMPR*WSTPI+WI*WSTPR                                           APP18800
  640 THETA=-TWOPI/FLOAT(IFACT(IF))                                     APP18810
      IF(ISIGN)650,645,645                                              APP18820
  645 THETA=-THETA                                                      APP18830
  650 WSTPR=COS(THETA)                                                  APP18840
      WSTPI=SIN(THETA)                                                  APP18850
      J2RNG=IFP1*(1+IFACT(IF)/2)                                        APP18860
      DO 695 I1=1,I1RNG,2                                               APP18870
      DO 695 I3=I1,NTOT,NP2                                             APP18880
      J2MAX=I3+J2RNG-IFP1                                               APP18890
      DO 690 J2=I3,J2MAX,IFP1                                           APP18900
      J1MAX=J2+IFP1-NP1                                                 APP18910
      DO 680 J1=J2,J1MAX,NP1                                            APP18920
      J3MAX=J1+NP2-IFP2                                                 APP18930
      DO 680 J3=J1,J3MAX,IFP2                                           APP18940
      JMIN=J3-J2+I3                                                     APP18950
      JMAX=JMIN+IFP2-IFP1                                               APP18960
      I=1+(J3-I3)/NP1HF                                                 APP18970
      IF(J2-I3)655,655,665                                              APP18980
  655 SUMR=0.                                                           APP18990
      SUMI=0.                                                           APP19000
      DO 660 J=JMIN,JMAX,IFP1                                           APP19010
      SUMR=SUMR+DATA(J)                                                 APP19020
  660 SUMI=SUMI+DATA(J+1)                                               APP19030
      WORK(I)=SUMR                                                      APP19040
      WORK(I+1)=SUMI                                                    APP19050
      GO TO 680                                                         APP19060
  665 ICONJ=1+(IFP2-2*J2+I3+J3)/NP1HF                                   APP19070
      J=JMAX                                                            APP19080
      SUMR=DATA(J)                                                      APP19090
      SUMI=DATA(J+1)                                                    APP19100
      OLDSR=0.                                                          APP19110
      OLDSI=0.                                                          APP19120
      J=J-IFP1                                                          APP19130
  670 TEMPR=SUMR                                                        APP19140
      SUMR=TWOWR*SUMR-OLDSR+DATA(J)                                     APP19150
      TEMPI=SUMI                                                        APP19160
      SUMI=TWOWR*SUMI-OLDSI+DATA(J+1)                                   APP19170
      OLDSR=TEMPR                                                       APP19180
      OLDSI=TEMPI                                                       APP19190
      J=J-IFP1                                                          APP19200
      IF(J-JMIN)675,675,670                                             APP19210
```

```
  675 TEMPR=WR*SUMR-OLDSR+DATA(J)                                   APP19220
      TEMPI=WI*SUMI                                                 APP19230
      WORK(I)=TEMPR-TEMPI                                           APP19240
      WORK(ICONJ)=TEMPR+TEMPI                                       APP19250
      TEMPR=WR*SUMI-OLDSI+DATA(J+1)                                 APP19260
      TEMPI=WI*SUMR                                                 APP19270
      WORK(I+1)=TEMPR+TEMPI                                         APP19280
      WORK(ICONJ+1)=TEMPR-TEMPI                                     APP19290
  680 CONTINUE                                                      APP19300
      IF(J2-I3)685,685,686                                          APP19310
  685 WR=WSTPR                                                      APP19320
      WI=WSTPI                                                      APP19330
      GO TO 690                                                     APP19340
  686 TEMPR=WR                                                      APP19350
      WR=WR*WSTPR-WI*WSTPI                                          APP19360
      WI=TEMPR*WSTPI+WI*WSTPR                                       APP19370
  690 TWOWR=WR+WR                                                   APP19380
      I=1                                                           APP19390
      I2MAX=I3+NP2-NP1                                              APP19400
      DO 695 I2=I3,I2MAX,NP1                                        APP19410
      DATA(I2)=WORK(I)                                              APP19420
      DATA(I2+1)=WORK(I+1)                                          APP19430
  695 I=I+2                                                         APP19440
      IF=IF+1                                                       APP19450
      IFP1=IFP2                                                     APP19460
      IF(IFP1-NP2)610,700,700                                       APP19470
C                                                                   APP19480
C     COMPLETE A REAL TRANSFORM IN THE 1ST DIMENSION, N EVEN, BY CON- APP19490
C     JUGATE SYMMETRIES.                                            APP19500
C                                                                   APP19510
  700 GO TO (900,800,900,701),ICASE                                 APP19520
  701 NHALF=N                                                       APP19530
      N=N+N                                                         APP19540
      THETA=-TWOPI/FLOAT(N)                                         APP19550
      IF(ISIGN)703,702,702                                          APP19560
  702 THETA=-THETA                                                  APP19570
  703 WSTPR=COS(THETA)                                              APP19580
      WSTPI=SIN(THETA)                                              APP19590
      WR=WSTPR                                                      APP19600
      WI=WSTPI                                                      APP19610
      IMIN=3                                                        APP19620
      JMIN=2*NHALF-1                                                APP19630
      GO TO 725                                                     APP19640
  710 J=JMIN                                                        APP19650
      DO 720 I=IMIN,NTOT,NP2                                        APP19660
      SUMR=(DATA(I)+DATA(J))/2.                                     APP19670
      SUMI=(DATA(I+1)+DATA(J+1))/2.                                 APP19680
      DIFR=(DATA(I)-DATA(J))/2.                                     APP19690
```

```
      DIFI=(DATA(I+1)-DATA(J+1))/2.                                     APP19700
      TEMPR=WR*SUMI+WI*DIFR                                             APP19710
      TEMPI=WI*SUMI-WR*DIFR                                             APP19720
      DATA(I)=SUMR+TEMPR                                                APP19730
      DATA(I+1)=DIFI+TEMPI                                              APP19740
      DATA(J)=SUMR-TEMPR                                                APP19750
      DATA(J+1)=-DIFI+TEMPI                                             APP19760
  720 J=J+NP2                                                           APP19770
      IMIN=IMIN+2                                                       APP19780
      JMIN=JMIN-2                                                       APP19790
      TEMPR=WR                                                          APP19800
      WR=WR*WSTPR-WI*WSTPI                                              APP19810
      WI=TEMPR*WSTPI+WI*WSTPR                                           APP19820
  725 IF(IMIN-JMIN)710,730,740                                          APP19830
  730 IF(ISIGN)731,740,740                                              APP19840
  731 DO 735 I=IMIN,NTOT,NP2                                            APP19850
  735 DATA(I+1)=-DATA(I+1)                                              APP19860
  740 NP2=NP2+NP2                                                       APP19870
      NTOT=NTOT+NTOT                                                    APP19880
      J=NTOT+1                                                          APP19890
      IMAX=NTOT/2+1                                                     APP19900
  745 IMIN=IMAX-2*NHALF                                                 APP19910
      I=IMIN                                                            APP19920
      GO TO 755                                                         APP19930
  750 DATA(J)=DATA(I)                                                   APP19940
      DATA(J+1)=-DATA(I+1)                                              APP19950
  755 I=I+2                                                             APP19960
      J=J-2                                                             APP19970
      IF(I-IMAX)750,760,760                                             APP19980
  760 DATA(J)=DATA(IMIN)-DATA(IMIN+1)                                   APP19990
      DATA(J+1)=0.                                                      APP20000
      IF(I-J)770,780,780                                                APP20010
  765 DATA(J)=DATA(I)                                                   APP20020
  770 I=I-2                                                             APP20030
      J=J-2                                                             APP20040
      IF(I-IMIN)775,775,765                                             APP20050
  775 DATA(J)=DATA(IMIN)+DATA(IMIN+1)                                   APP20060
      DATA(J+1)=0.                                                      APP20070
      IMAX=IMIN                                                         APP20080
      GO TO 745                                                         APP20090
  780 DATA(1)=DATA(1)+DATA(2)                                           APP20100
      DATA(2)=0.                                                        APP20110
      GO TO 900                                                         APP20120
C                                                                       APP20130
C     COMPLETE A REAL TRANSFORM FOR THE 2ND OR 3RD DIMENSION BY         APP20140
C     CONJUGATE SYMMETRIES.                                             APP20150
C                                                                       APP20160
  800 IF(I1RNG-NP1)805,900,900                                          APP20170
```

```
 805   DO 860 I3=1,NTOT,NP2                                      APP20180
       I2MAX=I3+NP2-NP1                                          APP20190
       DO 860 I2=I3,I2MAX,NP1                                    APP20200
       IMIN=I2+I1RNG                                             APP20210
       IMAX=I2+NP1-2                                             APP20220
       JMAX=2*I3+NP1-IMIN                                        APP20230
       IF(I2-I3)820,820,810                                      APP20240
 810   JMAX=JMAX+NP2                                             APP20250
 820   IF(IDIM-2)850,850,830                                     APP20260
 830   J=JMAX*NPO                                                APP20270
       DO 840 I=IMIN,IMAX,2                                      APP20280
       DATA(I)=DATA(J)                                           APP20290
       DATA(I+1)=-DATA(J+1)                                      APP20300
 840   J=J-2                                                     APP20310
 850   J=JMAX                                                    APP20320
       DO 860 I=IMIN,IMAX,NPO                                    APP20330
       DATA(I)=DATA(J)                                           APP20340
       DATA(I+1)=-CATA(J+1)                                      APP20350
 860   J=J-NPO                                                   APP20360
C                                                                APP20370
C      END OF LOOP ON EACH DIMENSION                             APP20380
C                                                                APP20390
 900   NPO=NP1                                                   APP20400
       NP1=NP2                                                   APP20410
 910   NPREV=N                                                   APP20420
 920   RETURN                                                    APP20430
       END                                                       APP20440
/*                                                               APP20450
//GO.SYSIN DD *                                                  APP20460
LA MESA VILLAGE, 13 MAY 83                                       APP20470
COIL ANTENNA AMP IN NT                                           APP20480
LA MESA VILLAGE, 13 MAY 83                                       APP20490
ASQ-81 AMP IN NT                                                 APP20500
LA MESA VILLAGE, 13 MAY 83                                       APP20510
FLUXGATE AMP IN NT                                               APP20520
LA MESA VILLAGE, 13 MAY 83                                       APP20530
PROGRAM OUTPUT IN NT                                             APP20540
LA MESA VILLAGE, 13 MAY 83                                       APP20550
RAW COIL TIME SERIES IN VOLTS                                    APP20560
LA MESA VILLAGE, 13 MAY 83                                       APP20570
COIL ANTENNA AMP IN NT                                           APP20580
LA MESA VILLAGE, 13 MAY 83                                       APP20590
ASQ81 AMP IN NT                                                  APP20600
LA MESA VILLAGE, 13 MAY 83                                       APP20610
FLUXGATE AMP IN NT                                               APP20620
LA MESA VILLAGE, 13 MAY 83                                       APP20630
PROGRAM OUTPUT IN NT                                             APP20640
LA MESA VILLAGE, 13 MAY 83                                       APP20650
```

```
RAW COIL TIME SERIES IN VOLTS                                          APP20660
/*                                                                     APP20670
//GO.FT20F001 DD UNIT=3400-4,VOL=SER=MIKE1,DISP=(OLD,KEEP),            APP20680
//             LABEL=(1,NL,,IN),                                       APP20690
//             DCB=(RECFM=FB,LRECL=32,BLKSIZE=512,DEN=2)               APP20700
//GO.SYSDUMP DD SYSOUT=A,OUTLIM=65000                                  APP20710
```

LIST OF REFERENCES

1. Air Force Geophysics Laboratory, L. G. Hanscom Field, Bedford, Massachusetts, Report AFGL-79-0192, Proceedings of the Air Force Geophysics Laboratory Workshop on Geomagnetism: April 6-7, 1979, Edited by R. C. Sagalyn, R. O. Hutchinson, and S. Gussenhoven, p. 47, 1979.

2. Jacobs, J. A., Geomagnetic Micropulsations, Springer-Verlag, New York, 1970.

3. Air Force Cambridge Research Laboratories, L. G. Hanscom Field, Bedford, Massachusetts, Report AFCRL-72-0570, The Geomagnetic Field, by David J. Knecht, 1972.

4. Naval Oceanographic Office, Washington, D. C., Technical Report TR-218, The Influence of the Natural Environment on MAD Operations, by James A. Brennan and Thomas M. Davis, September 1969.

5. Chilton, F., Wood, L., and Buntzen, R., "Electric and Magnetic Sensing Systems: Applications," chapter 10 of Applications of Remore Sensing to Ocean Surveillance, AGARD Lecture Series No. 88, North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, London, 1977.

6. Naval Air Systems Command NAVAIR 16-30ASQ81-500,Preliminary Manual Interim Field Engineering Maintenance Instructions with Illustrated Parts Breakdown, Magnetic Detecting Set AN/ASQ-81(V)2, 1 November 1971.

7. Oppenheim, Alan V., and Schafer, Ronald W., Digital Signal Processing, Prentice-Hall, Inc., New Jersey, 1975.

8. Jury, E. I., Theory and Application of the z-Transform Method, John Wiley and Sons, Inc., New York, 1964.

9. Naval Air Development Center, Warminster, Pennsylvania, Report NADC-EL-47-50, Magnetic Airborne Detection Frequency Responses, by J. E. Anderson, 1949.

10. Stevens, Kurt B., Evaluation of the Naval Postgraduate School Geomagnetic Sensing System and Software, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1983.

BIBLIOGRAPHY

Bird, G. J. A., Design of Continuous and Digital Electronic Systems, McGraw-Hill Book Company Limited, 1980.

Close, C. M., The Analysis of Linear Circuits,Harcourt, Brace and World, Inc., 1966.

Humpherys, D. S., The Analysis, Design, and Synthesis of Electrical Filters, Prentice-Hall, Inc., 1970.

Jury, E. I., Theory and Application of the Z-Transform Method, John Wiley and Sons, Inc., 1964.

NAVAIR 16-30ASQ81-500, Preliminary Manual, Interim Field Engineering Maintenance Instructions with Illustrated Parts Breakdown, Magnetic Detecting Set AN/ASQ-81(V)2, (Texas Instruments, Inc.), 1 November 1971.

Oppenheim, A. V., and Schafer, R. W., Digital Signal Processing, Prentice-Hall, 1975.

Rabiner, L. R., and Gold, B., Theory and Application of Digital Signal Processing, Prentice-Hall, Inc., 1975.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA. 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA. 93943	2
3.	Chairman ASW Academic Group, Code 71 Naval Postgraduate School Monterey, CA. 93943	1
4.	Dr. Otto Heinz, Code 61Hz Department of Physics Naval Postgraduate School Monterey, CA. 93943	2
5.	Dr. Andrew R. Ochadlick, Jr., Code 61Oc Department of Physics Naval Postgraduate School Monterey, CA. 93943	2
6.	Dr. Paul Moose, Code 62Me Department of Electrical Engineering Naval Postgraduate School Monterey, CA. 93943	1
7.	LT Michael C. Huete, USN Destroyer Squadron TWENTY-ONE FPO San Francisco, CA. 96601	2
8.	Mr. John Shannon Naval Air Development Center (Code 3012) Warminster, PA. 18974	1
9.	Mr. Edward Yannuzzi Naval Air Development Center (Code 30) Warminster, PA. 18974	1

10. Chief of Naval Research
Department of the Navy
Attn: Code 100C1 1
Code 425 GG 1
Code 460 1
Code 464 1
Code 480 1
800 North Quincy Street
Arlington, VA. 22217

11. Mr. William Andahazy 1
Naval Ship Research & Development Center
Annapolis Laboratory
Annapolis, MD. 21402

12. Naval Air Systems Command 1
Attn: Mr. Barry Dillon
Code AIR-340J
Washington, DC 20361

13. Fleet Tactical Library 1
P.O. Box 1042
Silver Springs, MD. 20910

202558

Thesis
H8535 Huete
c.1
A digital filter
representation of the
ASQ-81 magnetometer.